



DECKBLATT

EU 036.16	Projekt	PSP-Element	Obj. Kenn.	Aufgabe	UA	Lfd. Nr.	Rev
	N A A N	N N N N N N N N N N	N N N N N N	X A A X X	A A	N N N N	N N
	9K	324.34		EG	RB	0002	00

Titel der Unterlage: Emos Manual, Version 2.1: Programmpaket zur Modellierung der Radionuklidfreisetzung aus Endlagern für angenommene Szenarien	Seite:
	I.
	Stand:
	15.3.86

Ersteller:	Textnummer:
GSP	

Stempelfeld:

PSP-Element TP...9K/21285	zu Plan-Kapitel: 3.9
---------------------------	----------------------

	PL	PL
	24.3.86	24.3.86
	Freigabe für Behörden	Freigabe im Projekt

Diese Unterlage unterliegt samt Inhalt dem Schutz des Urheberrechts sowie der Pflicht zur vertraulichen Behandlung auch bei Beförderung und Vernichtung und darf vom Empfänger nur auftragsbezogen genutzt, vervielfältigt und Dritten zugänglich gemacht werden. Eine andere Verwendung und Weitergabe bedarf der ausdrücklichen Zustimmung der PTB.

Revisionsblatt



EU 036.16	Projekt	PSP-Element	Obj. Kenn.	Aufgabe	UA	Lfd. Nr.	Rev.
	N A A N	N N N N N N N N N N	N N N N N N	X A A X X	A A	N N N N	N N
	9K	324.34		EG	RB	0002	00

Titel der Unterlage: Emos Manual, Version 2.1: Programmpaket zur Modellierung der Radionuklidfreisetzung aus Endlagern für angenommene Szenarien	Seite: II.
	Stand: 15.3.86

Rev.	Revisionsst. Datum	verant. Stelle	Gegenzeichn. Name	rev. Seite	Kat. *)	Erläuterung der Revision

*) Kategorie R = redaktionelle Korrektur
 Kategorie V = verdeutlichende Verbesserung
 Kategorie S = substantielle Änderung
 Mindestens bei der Kategorie S müssen Erläuterungen angegeben werden.

E M O S

PROGRAMMPAKET ZUR MODELLIERUNG
DER RADIONUKLIDFREISETZUNG AUS
ENDLAGERN FÜR ANGENOMMENE
SZENARIEN

E M O S 2

VERSION 2.1 VOM 15. MÄRZ 1986

Gesellschaft für Strahlen- und Umweltforschung mbH München
Institut für Tief Lagerung
Braunschweig

INHALTSVERZEICHNIS

1 EINLEITUNG

- 1.1 Rechenprogramme für die Langzeitsicherheit
- 1.2 Aufgaben von EMOS
- 1.3 Entwicklungsgeschichte
- 1.4 Weiterentwicklung
- 1.5 Dokumentationsweise

2 ÜBERSICHT ÜBER DAS PROGRAMMSYSTEM

- 2.1 Ansteuerung von Moduln (EMOS2)
- 2.2 Dateneingabemöglichkeiten

TEIL A: PROGRAMMBESCHREIBUNGEN

A.1 SERVICE ROUTINEN

- A 1.1 Auflistung der Datenbibliothek (PERMLIS)
- A 1.2 Ansprechen der Datenbibliothek (OPENFIL)
- A 1.3 Hilfsroutine für Module (HRANF)
- A 1.4 Hilfsfunktionen bei der Ausdrucksteuerung (NOUTBAR)

A 2 RADIONUKLIDFREISETZUNG AUS EINEM GRUBENGEBÄUDE

- A 2.1 Programmkonzept für Freisetzungsrechnungen
- A 2.2 Programmstruktur (REPOS)
- A 2.3 Ein-/Ausgabe von Steuerdaten (STEUDA)
- A 2.4 Ein-/Ausgabe modellspezifischer Eingangsdaten (DALEKO)

A 2.4.1	Nuklidspezifische Daten	
A 2.4.1.1	Eingabe nuklidspezifischer Daten	(NUDAIN)
A 2.4.1.2	Ausgabe nuklidspezifischer Daten	(DRUNUDA)
A 2.4.2	Elementspezifische Daten	
A 2.4.2.1	Eingabe elementspezifischer Daten	(ELDAIN)
A 2.4.2.2	Ausgabe elementspezifischer Daten	(DRUELDA)
A 2.4.3	Ein-/Ausgabe von Strukturdaten	(SDATIN)
A 2.4.4	Barrierenspezifische Daten	(BDATIN)
A 2.4.4.1	Ausgabe globaler Barrierendaten	(PRTGLB)
A 2.4.4.2	Eingabe barrierenspezifischer Daten	(BADAIN)
A 2.4.4.3	Überprüfung einer Barriere	(PRBAST)
A 2.4.4.4	Übernahme für eine Barriere	(BMIN)
A 2.4.4.5	Ausgabe barrierenspezifischer Daten	(DRUBADA)
A 2.4.5	Kodierung der Strukturdaten	(KSEBUE)
A 2.4.6.	Abfallspezifische Daten	(ADATIN)
A 2.4.6.1	Eingabe abfallspezifischer Daten	(ABDAIN)
A 2.4.6.2	Zwischenlagerung der Abfallgebinde	(ZWLAG)
A 2.4.6.3	Ausgabe abfallspezifischer Daten	(DRUABDA)
A 2.4.7	Inventar des Barrierensystems	(BARINV)
A 2.5	Berechnung der Freisetzung	(STEBA)
A 2.5.1	Schrittweitensteuerung	
A 2.5.2	Aufruf von Barrierenmodellen	(BMOD)
A 2.5.3	Barrierenmodelle	
A 2.5.3.1	Elementspezifische Mobilisierung	(MOBZ6)
A 2.5.3.2	Volumenquelle	(VQUELLE1)
A 2.5.4	Einzeleffekte	
A 2.5.4.1	Radioaktiver Zerfall und Bilanzierung	(RAZE)
A 2.5.4.2	Sorption und Ausfällung	(SORPRP2)

- A 2.5.5 Hilfsroutinen
- A 2.5.5.1 Umrechnung von Aktivitäten (ELMAKO)
- A 2.5.5.2 Barrieren- oder effektbezogene Zeitschritt-
steuerung (SETDTMA)
- A 2.5.6 Ausgabe während der Rechnung
- A 2.5.7 Ausgabe am Ende der Rechnung
- A 2.5.7.1 Ausgabe des Zuflußverlaufs (ZEITEN)
- A 2.5.7.2 Ausgabe von Tabellen mit Löslichkeitszeit-
punkten und maximalen Elementkonzentrationen (MAXEL)
- A 2.5.7.3 Ausgabe von Freisetzungsmengen (DRUFM)

A 3 COMMON-BLÖCKE

- A 3.1 Bezeichnung von Variablen und Feldern
- A 3.2 Dimensionierungsparameter
- A 3.3 Beschreibung der COMMON-Blöcke

TEIL B: STRUKTUREN UND HANDHABUNG DER DATENBIBLIOTHEK

- B 1 DATENBIBLIOTHEK
- B 1.1 Verwaltung der Datenbibliothek
- B 1.2 Aufbau des Index-Files
- B 1.3 Aufbau der Daten-Files
- B 1.3.1 Nuklidspezifische Daten
- B 1.3.2 Elementspezifische Daten
- B 1.3.3 Abfallspezifische Daten
- B 1.3.4 Barrierenspezifische Daten
- B 1.4 Handhabung der Datenbibliothek (JUNK)

TEIL C: HANDHABUNG DER PROGRAMME

C 1 ÜBERSICHT ÜBER PROGRAMMDATEIEN

C 2 PROZEDUREN

C 2.1	Ergänzung der Objektbibliothek	(ADDBIB)
C 2.2	Erzeugung einer Objektbibliothek	(NEWBIB)
C 2.3	Binden des Programms	(LINKEMOS2)
C 2.4	Durchführung von Rechenläufen	(RUNEMOS2)

TEIL D: EINGABEBESCHREIBUNG

D 1 ÜBERSICHT ÜBER DAS JOB-INPUT-FILE

D 2 EINGABE DER DATEN FÜR DAS JOB-INPUT-FILE

D 2.1 Datenfiles für den Modul

D 2.2 Steuerparameter zur Datenausgabe

D 2.2.1 Auswahl von Barrieren und Nukliden

D 2.2.2 Steuerung der Ausgabe der Eingabedaten

D 2.2.3 Steuerung der Ausgabe von Zwischenergebnissen

D 2.2.4 Steuerung der Ergebnisausgabe am Ende der Rechnung

D 2.2.5 Steuerung der Plotausgabe

D 2.3 Parameter zum Zeitrahmen und zur Schrittweitensteuerung

D 2.4 Struktur des Barrierensystems

D 2.5 Barrierendaten

D 2.5.1 Globale Barrierendaten

D 2.5.2 Spezifische Barrierendaten

D 2.6 Spezifizierung der Abfallmixturen

1 EINLEITUNG

Für die Endlagerung radioaktiver Abfälle werden geeignete geologische Formationen im tiefen Untergrund betrachtet. Die Hauptaufgabe eines solchen Endlagers ist die sichere Isolierung der eingelagerten Abfälle von der Biosphäre. Da hierbei lange Zeiträume zu betrachten sind, ist eine absolute Isolierung nicht unter allen Bedingungen zu gewährleisten. Bei einer unerwünschten Entwicklung des Endlagersystems, aber möglicherweise auch bei einer normalen Entwicklung kann eine Kontamination der Biosphäre in der fernen Zukunft nicht ausgeschlossen werden.

Die verschiedenen Entwicklungsmöglichkeiten, die eine Kontamination der Biosphäre verursachen können, werden zu Beginn der Analyse der Langzeitsicherheit eines Endlagers als Szenario definiert. Für solche Szenarien müssen die radiologischen Konsequenzen für die dann lebenden Menschen ermittelt werden. Wegen des ausgedehnten Betrachtungszeitraums kann dies nicht durch Experimente, sondern nur durch Modellrechnungen geschehen. Da es sich bei dem Gesamtsystem des Endlagers hinsichtlich der Radionuklidenausbreitung um komplexe Vorgänge handelt, sind hierfür Rechenprogramme erforderlich.

1.1 RECHENPROGRAMME FÜR DIE LANGZEITSICHERHEIT

Die Aufgabe der Rechenprogramme für die Langzeitsicherheit ist letztendlich die Berechnung der Strahlenexposition eines Individuums als Funktion der Zeit für ein vorgegebenes Szenario. Da die Strahlenexposition in der Biosphäre von einer Vielzahl physikalischer und chemischer Prozesse abhängt, sind unterschiedlichste Rechenprogramme erforderlich. Diese lassen sich in zwei Hauptkategorien unterteilen:

- Programme zur detaillierten Untersuchung von Einzeleffekten
- Programme zur eigentlichen Berechnung der Nuklidausbreitung

Durch die Untersuchung von Einzeleffekten werden Modellansätze und Daten bereitgestellt, die in der Nuklidausbreitung berücksichtigt werden müssen. Die Bereitstellung von Daten für die Einzeluntersuchungen und für die Nuklidausbreitung sowie der Nachweis der Gültigkeit von Modellansätzen erfolgt durch experimentelle Untersuchungen. Der geschilderte Sachverhalt ist in Abbildung 1-1 zusammen mit Beispielen für Einzeluntersuchungen dargestellt.

Die Programme zur Berechnung der Radionuklidausbreitung lassen sich bezüglich der drei Hauptkomponenten des Endlagersystems in

- Grubengebäude oder Nahbereich,
- Deckgebirge oder Fernbereich und
- Biosphäre

unterteilen. Da die Vorgänge in den drei Hauptkomponenten oftmals gänzlich unterschiedlich sind, existieren entsprechend eigenständige Programmteile, die zur Berechnung der Strahlenexposition nacheinander durchgerechnet werden müssen.

Die möglichen Vorgänge in den Hauptkomponenten eines Endlagersystems sind abhängig von der Art der Endlagerformation, dem technischen Einlagerungskonzept und von dem betrachteten Szenario. Da außerdem die Modellansätze für die physikalischen und chemischen Vorgänge sich mit fortschreitendem Entwicklungsstand ändern, ist eine hohe Flexibilität in den Rechenprogrammen erforderlich.

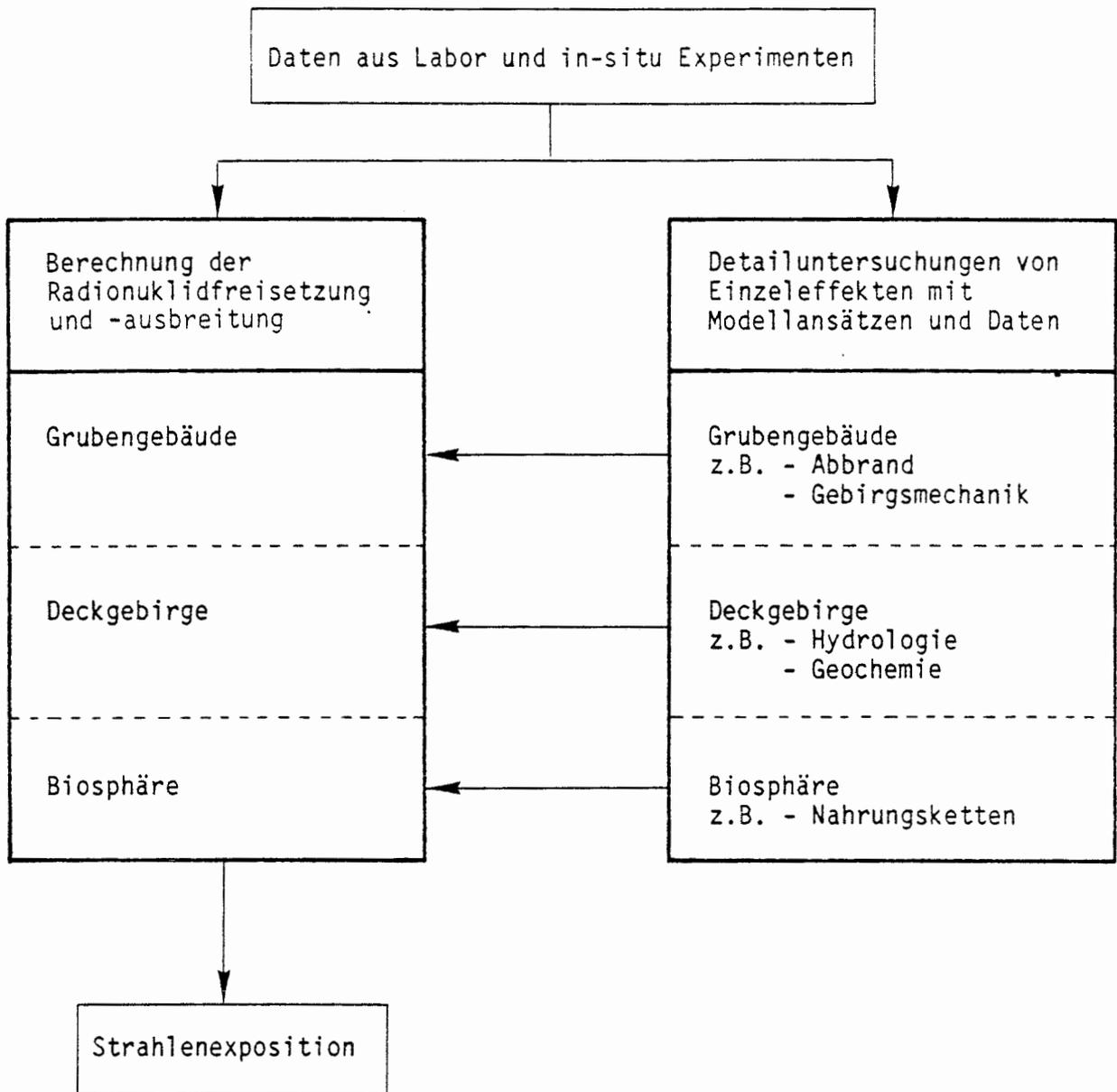


Abb. 1-1: Aufgabenbereiche in einer Langzeitsicherheitsanalyse eines Endlagers

1.2 AUFGABEN VON EMOS

Der Aufbau des Rechenprogramms EMOS erlaubt die Berechnung der Nuklid-
ausbreitung durch die drei Hauptkomponenten des Endlagersystems. Die
Aufgabe des Programms EMOS ist die Ansteuerung und Datenversorgung von
Programmmoduln, die jeweils die Nuklidausbreitung in einer Hauptkompo-
nente berechnen. Bei einer sukzessiven Durchrechnung aller drei
Hauptkomponenten ist außerdem der Datentransfer zwischen den Moduln
sicherzustellen und die Aufarbeitung und Darstellung von Rechenergeb-
nissen zu gewährleisten.

In der aktuellen Version von EMOS sind nur Moduln für das Grubenge-
bäude verfügbar. Ein Modul für das Deckgebirge in Form eines eindimen-
sionalen Transportprogramms wird in naher Zukunft zur Verfügung stehen.
Die beiden Moduln für das Grubengebäude behandeln einmal ein Endlager
im Salz und einmal ein Endlager im Eisenerz. Die vorliegende Dokumen-
tation umfaßt nur das Steuerprogramm EMOS und das Modul für ein Gru-
bengebäude in einer Eisenerzformation (REPOS).

1.3 ENTWICKLUNGSGESCHICHTE

Das Steuerprogramm EMOS und das Modul für ein Grubengebäude im Salz
wurden in den Jahren 1981 bis 1984 im Rahmen des Projektes "Sicher-
heitsstudien Entsorgung" (PSE) an der Technischen Universität Berlin
entwickelt. Der Schwerpunkt lag dabei auf der Entwicklung und Erpro-
bung von Modellansätzen für die Vorgänge in einem Grubengebäude eines
Salzstocks. Die programmtechnische Realisierung wurde auf einer CD-
Rechenanlage unter Verwendung von Overlay-Strukturen und dynamisch
adressierbaren Arbeitsspeichern vollzogen. Eine Dokumentation dieser
ersten Version des Programmes EMOS liegt nicht vor.

Seit Beginn des Jahres 1985 wurden die Entwicklungsarbeiten an dem Institut für Tieflagerung der Gesellschaft für Strahlen- und Umweltforschung mbH auf einer DEC-Rechenanlage weitergeführt. Schwerpunkt der programmtechnischen Weiterentwicklung ist zunächst die Neustrukturierung des gesamten Programms und die Umstellung auf ein virtuelles Betriebssystem. Bei einem virtuellen Betriebssystem ist die Verwendung von Overlay-Strukturen und dynamisch adressierbaren Arbeitsspeichern durch den Anwender nicht mehr erforderlich.

Die Umstellung auf ein virtuelles Betriebssystem wurde Mitte 1985 abgeschlossen. Damit verbunden war eine vollständige Umstellung der Verwaltung der Datenbibliothek sowie die Datenspeicherung im Programm. Gleichzeitig mit der Umstellung der Datenspeicherung wurden die Programmteile zur Datenübernahme und Datenaufbereitung (DALEKO und zugehörige Unterprogramme) neu strukturiert und programmiert. Eine Neustrukturierung der Programmteile zur Durchführung der zeitdiskreten Rechnung (STEBA und zugehörige Unterprogramme) soll in naher Zukunft erfolgen.

Das Modul für ein Grubengebäude im Salz war auf ein Grubengebäude im Erz nicht unmittelbar übertragbar. Die Modifikationen führten zur Entstehung eines neuen Moduls, in dem jedoch die wesentlichen Teile der Datenübernahme und der Steuerung der zeitdiskreten Rechnung erhalten geblieben sind. Die Entwicklung des Moduls REPOS wurde Ende 1985 abgeschlossen.

1.4 WEITERENTWICKLUNG

Die Neustrukturierung der Programmteile zur Durchführung der zeitdiskreten Rechnung wird demnächst vorbereitet und soll bis Ende 1986 abgeschlossen sein. Als Abschluß einer programmtechnischen Entwicklung über sechs Jahre liegen dann die Module zum Grubengebäude in einer einheitlichen und übersichtlichen Struktur vor.

Als Einstieg in die zweite Hauptkomponente des Endlagersystems ist die Implementation eines eindimensionalen Transportmodells bis Ende 1986 vorgesehen. Nach der Sammlung ausreichender Erfahrungen können dann mehrdimensionale Ausbreitungsmodelle integriert werden. Im Gegensatz zum Grubengebäude werden die Ausbreitungsprogramme zum Deckgebirge nicht selbst entwickelt, sondern fertig übernommen. Zur Integration in das Steuerprogramm EMOS sind jeweils programmtechnische Anpassungsarbeiten erforderlich.

Zur Berücksichtigung der Unsicherheiten in den Eingangsdaten der einzelnen Moduln soll das Steuerprogramm EMOS um einen Statistikteil erweitert werden. Bei Vorgabe von Verteilungsfunktionen der Eingangsparameter können dann im Rahmen einer Monte Carlo-Simulation Verteilungsfunktionen der Berechnungsergebnisse ermittelt werden. Diese Erweiterungsarbeiten sind für das Jahr 1987 geplant.

Die Ausgabe von Rechenergebnissen zur graphischen Darstellung und die Erzeugung von Plots geschah in der ersten EMOS-Version zusammen mit der Ergebniserzeugung in einem einzigen Rechenlauf. Bedingt durch die erforderlichen Zeiten für die Rechnungsdurchführung und die Zeichnungserstellung war dadurch die Herstellung inhaltsreicher Abbildungen ein zeitintensiver Prozess. Daher wurde Anfang 1986 mit Programmentwicklungen begonnen, die eine Ablage aller Rechenergebnisse auf einem Massenspeicher bei der Berechnung vorsieht und eine anschließende, interaktive Erstellung von Zeichnungen ermöglicht. Diese Entwicklungsarbeiten sollen Mitte 1986 abgeschlossen sein.

1.5 DOKUMENTATIONSWEISE

Die vorliegende Dokumentation beinhaltet das Steuerprogramm EMOS und die zugehörige Datenbibliothek sowie den Programmmodul REPOS für ein Grubengebäude in einer Erzformation. Die Programmteile zur Datenübernahme und Datenablage liegen dabei in der neustrukturierten Form (EMOS2) und die Programmteile zur Durchführung der zeitdiskreten Rechnung in der ursprünglichen Struktur (EMOS1) vor. Alle Programmteile sind in einheitlicher Form dokumentiert.

Die vorliegende Dokumentation enthält nach der Einführung und der Übersicht die folgenden Teile:

- Teil A: Programmbeschreibungen
- Teil B: Strukturen und Handhabung von Daten
- Teil C: Handhabung von Programmen
- Teil D: Eingabebeschreibungen

Die Reihenfolge der Programmbeschreibungen im Teil A orientiert sich an der Programmhierarchie und dem Ablauf des Programms. Die Beschreibung eines Programmteils ist in allen Fällen in der folgenden Art aufgebaut:

- beschreibender Text zu den Aufgaben des Programmteils, den zugrundeliegenden Gleichungen und zum Ablauf des Programmteils in groben Zügen,
- formale Übersicht über den Programmteil entsprechend den Stichpunkten in Tabelle 1-1,
- Ablaufdiagramme in grober Darstellung unter Berücksichtigung von Unterprogrammaufrufen.

EINZELPUNKT	ANGABE	BEMERKUNG
-----	-----	-----
Programmname	Name	-
Übergeordnetes Programm	Name	-
Länge des Programms	Zeilen, Bytes	Gesamtzahlen der Programmzeilen Speicherumfang des Rechencodes
Stand der Dokumentation	Datum	-
Kurzbeschreibung	Text	-
-----	-----	-----
Parameterliste	Variablennamen	-
Eingangsparameter	Namen, Bedeutung	für Berechnungen erforderlich
Durchgangsparameter	Namen, Bedeutung	für Unterprogrammaufrufe
Ausgangsparameter	Namen, Bedeutung	Ergebnisse von Berechnungen
-----	-----	-----
Common-Blöcke	-	-
lesender Zugriff	Block-, Variablennamen	Variablennamen nur in Einzelfällen
schreibender Zugriff	Block-, Variablennamen	Variablennamen nur in Einzelfällen
Zugriff beim Unter- programmaufruf	Block-, Variablennamen	Variablennamen nur in Einzelfällen
-----	-----	-----
Unterprogrammaufrufe	Programmname, Position, Parameterliste	die Position bezieht sich auf den Aufruf im Ablaufdiagramm
-----	-----	-----

Tab. 1-1: Einzelpunkte einer Übersicht über ein Programmteil

Neben der Beschreibung der Programmteile eines Moduls enthält Teil A ein Kapitel zur Beschreibung von Hilfsroutinen sowie der Struktur von COMMON-Blöcken.

In den Teilen B und C werden die Strukturen und Inhalte der Datenfiles und die Untergliederung der Programmdateien, sowie die Handhabung der Datenfiles und Programmdateien unter Verwendung entsprechender Prozeduren erläutert. In Teil C werden außerdem die Prozeduren zur Durchführung von Rechenläufen beschrieben. Die Eingabebeschreibung im Teil D beinhaltet die Dateneingabe für einen Rechenlaufs über das Job-Input-File.

2 ÜBERSICHT ÜBER DAS PROGRAMMSYSTEM

Das Programm EMOS organisiert die Berechnung der Nuklid- ausbreitung in den einzelnen Hauptkomponenten Grubengebäude, Deckgebirge und Biosphäre eines Endlagersystems. Die Möglichkeit zur Berechnung der Nuklid- ausbreitung in einzelnen oder mehreren Hauptkomponenten hängt von den verfügbaren Moduln für die einzelnen Komponenten ab.

2.1 ANSTEUERUNG VON MODULN (EMOS2)

EMOS2 ist das Hauptprogramm des Programmpaketes und dient im wesentlichen der Ansteuerung der Module. Die Ansteuerung eines Moduls wird durch Angabe eines Modulnamens im Job-Input-File ausgelöst. Gleichzeitig mit der Angabe eines Modulnamens werden Datenfiles benannt, die aus der Datenbibliothek zu übernehmen sind.

Neben der direkten Ansteuerung von Moduln wird durch das Steuerprogramm EMOS2 eine Auflistung der in der Datenbibliothek vorhandenen Datenfiles jeweils vor und nach dem Aufruf eines Moduls veranlaßt.

ÜBERSICHT ÜBER DAS PROGRAMM EMOS2

ÜBERGEORDNETES PROGRAMM : -
LÄNGE DES PROGRAMMS : 106 Zeilen
436 Bytes
STAND DER DOKUMENTATION : 15. März 1985

KURZBESCHREIBUNG:

Ansteuerung und Auflistung der Datenbibliothek und Aufruf einer Folge von Moduln

LESENDER ZUGRIFF AUF COMMON-BLÖCKE:

/ / : ZEIT

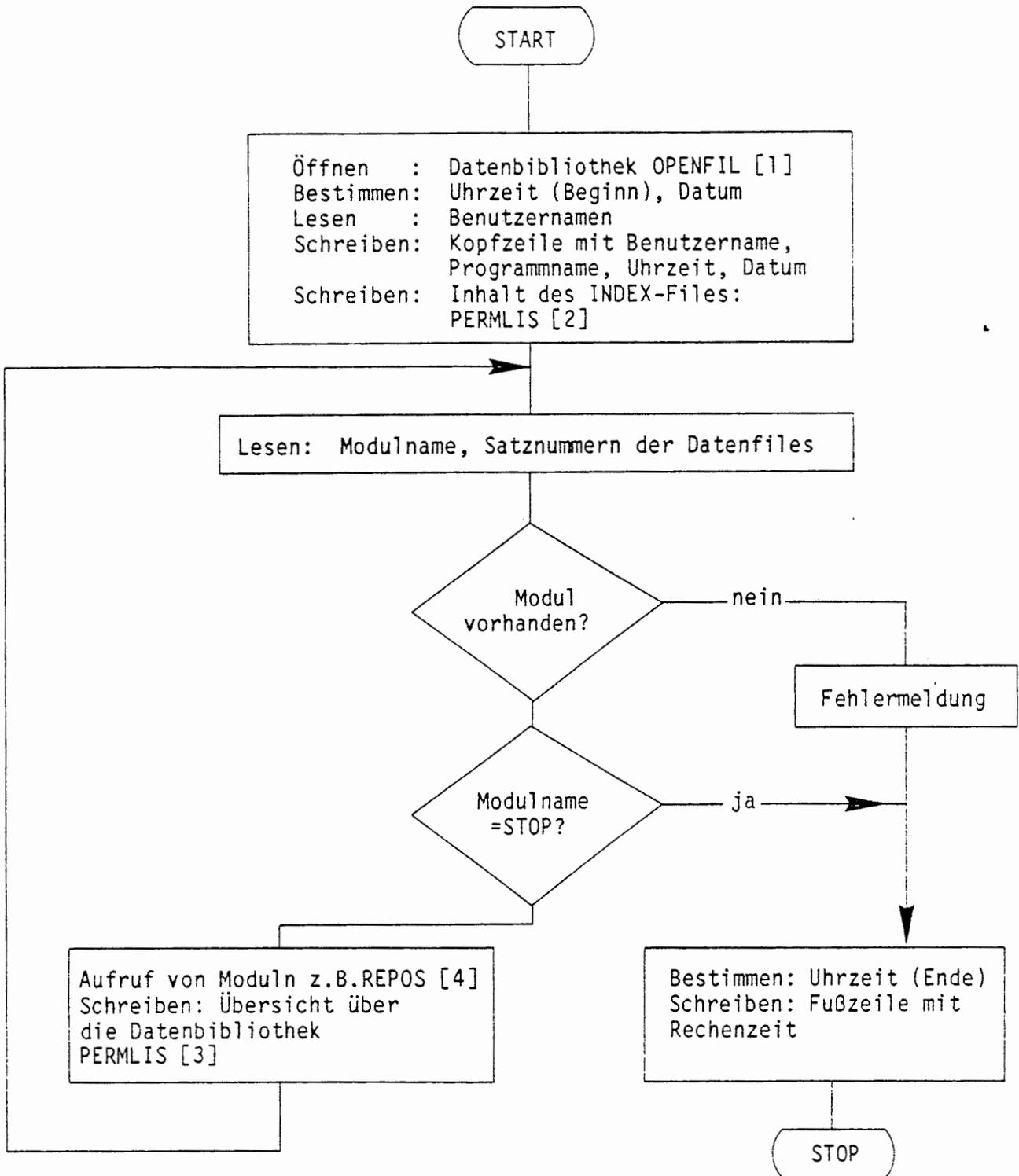
SCHREIBENDER ZUGRIFF AUF COMMON-BLÖCKE:

/ / : ZEIT, IST

UNTERPROGRAMMAUFRUFE:

- [1] OPENFIL (0, 'INDEX')
- [2] PERMLIS
- [3] PERMLIS
- [4] z.B. REPOS

ABLAUF DES PROGRAMMS EMOS2



2.2 DATENEINGABEMÖGLICHKEITEN

Die Dateneingabe in einen Programmmodul erfolgt über das Job- Input-File und aus der Datenbibliothek.

Die Dateneingabe für einen Modul über das Job-Input-File beinhaltet Steuerdaten für den zeitdiskreten Programmablauf und für die Ausgabe von Rechenergebnissen sowie modellspezifische Eingangsdaten für die berücksichtigten physikalischen und chemischen Teilmodelle. Die Dateneingabe über das Job-Input-File ist in der Eingabebeschreibung in Teil D erläutert.

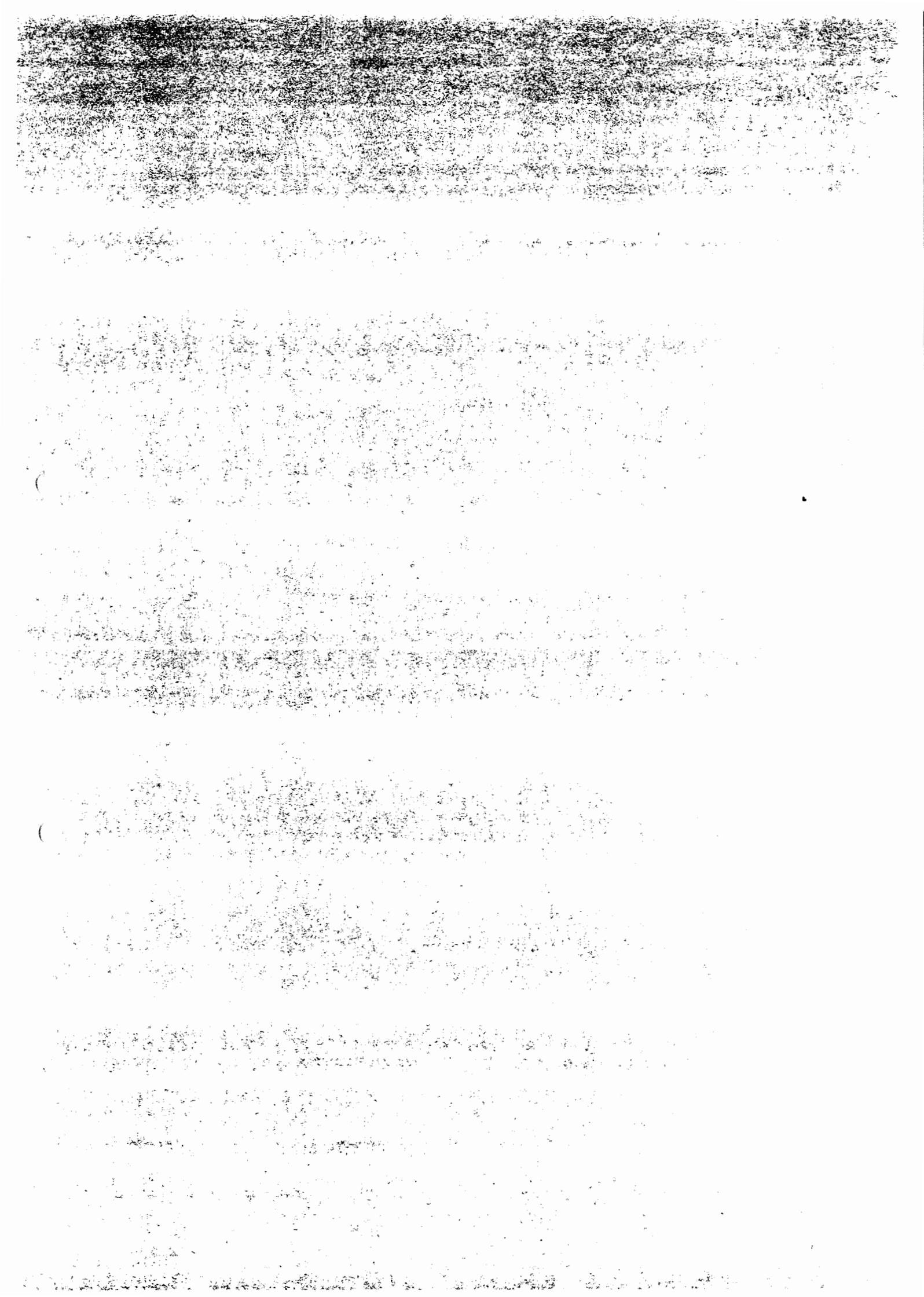
Die Eingabe größerer Datenmengen in ein Modul erfolgt von der Datenbibliothek. Die Ansteuerung eines Datenfiles der Bibliothek geschieht unmittelbar bei der Ansteuerung eines Moduls durch Angabe entsprechender Satznummern.

In der Datenbibliothek können maximal 99 Datenblöcke abgelegt werden. Die Einbringung von Datenblöcken sowie die Korrektur oder die Löschung von Datenblöcken geschieht über einen kontrollierten Zugriff, der durch eine Prozedur gesteuert wird. Dies wird im Teil B näher erläutert.

In der Datenbibliothek können vier unterschiedliche Datenstrukturen abgelegt werden. Diese enthalten jeweils unterschiedliche Datenarten

- nuklidspezifische Daten
- elementspezifische Daten
- abfallspezifische Daten
- barrierenspezifische Daten

Die jeweiligen Strukturen der Datenarten und die zugehörigen Formate werden bei der Beschreibung der Datenbibliothek erläutert.



TEIL A: PROGRAMMBESCHREIBUNGEN

A 1 SERVICE-ROUTINEN

Als Service-Routine werden solche Unterprogramme bezeichnet, die nicht direkt dem Aufgabengebiet eines einzelnen Moduls zugeordnet werden können.

A 1.1 AUFLISTUNG DER DATENBIBLIOTHEK (PERMLIS)

Zur Erstellung einer Übersicht über den Inhalt der Datenbibliothek wird das Index-File der Bibliothek gelesen und die Charakteristika der vorhandenen Datenfiles ausgedruckt. Gleichzeitig wird der Umfang eines Datenfiles in Blocks abgespeichert und derjenige der gesamten Bibliothek berechnet.

ÜBERSICHT ÜBER DAS PROGRAMM PERMLIS

ÜBERGEORDNETES PROGRAMM : EMOS2

LÄNGE DES PROGRAMMS : 53 Zeilen
289 Bytes

STAND DER DOKUMENTATION : 15. März 1986

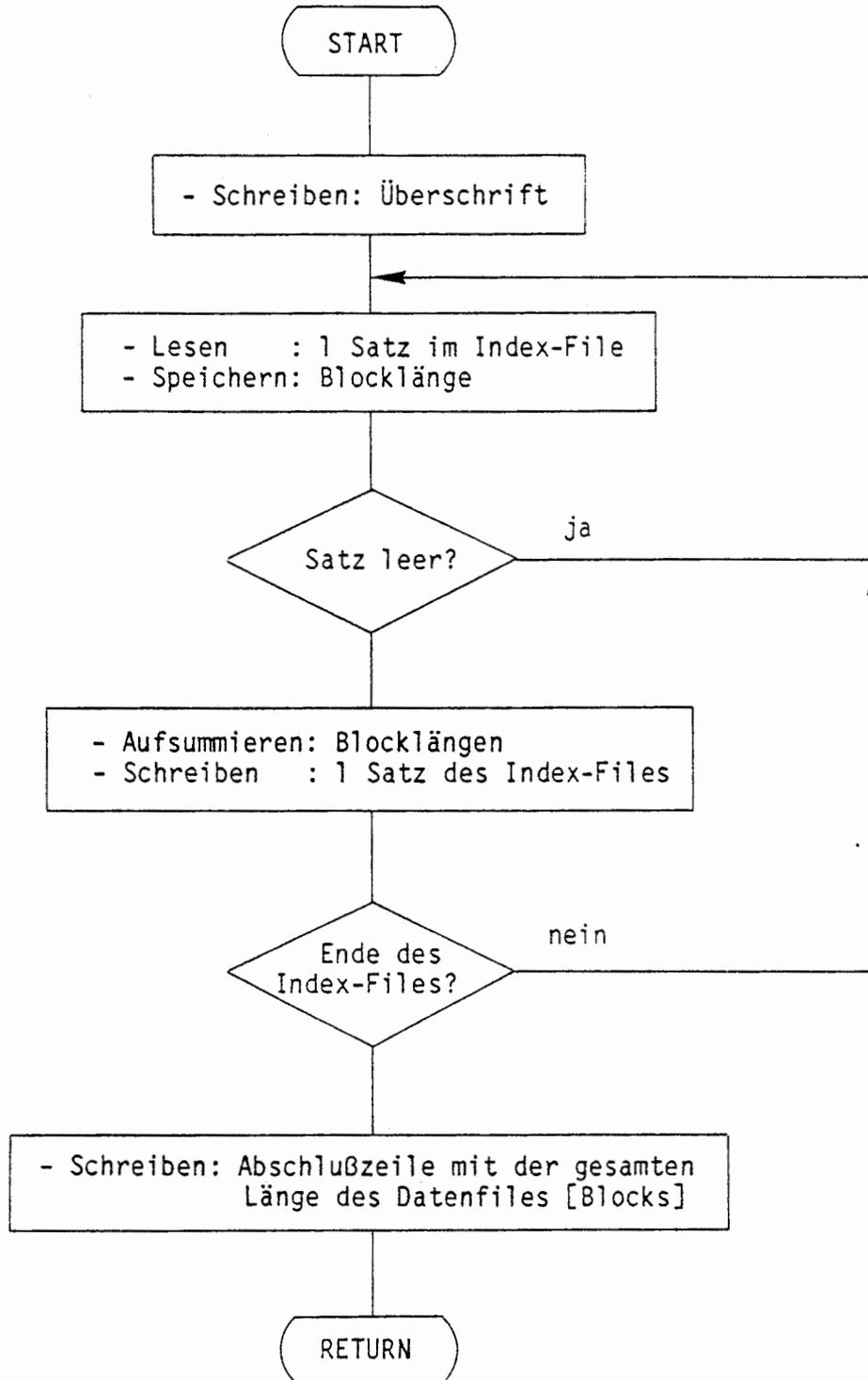
KURZBESCHREIBUNG:

Auflisten des Inhalts der Datenbibliothek

SCHREIBENDER ZUGRIFF AUF COMMON-BLÖCKE:

/ / : LFILE

ABLAUF DES PROGRAMMS PERMLIS



A 1.2 ANSPRECHEN DER DATENBIBLIOTHEK (OPENFIL)

Die Datenbibliothek besteht aus den einzelnen Datenfiles sowie einem Index-File zur Verwaltung der Bibliothek. Durch Aufruf der Routine OPENFIL bzw. des Entries CLOSEFIL werden die Files mit folgender UNIT-Nummer geöffnet bzw. wieder geschlossen

Index-File : UNIT = 10
Daten-Files : UNIT = 11

Zum Ansprechen des Index-Files muß der Eingangsparameter CTYPX mit 'Index' vereinbart werden. Zum Ansprechen eines Daten-Files muß neben einer Nummer für das Datenfile der Typ des Datenfiles eingegeben werden. Dieser wird dann mit dem Typ des Datenfiles in der Bibliothek verglichen.

ÜBERSICHT ÜBER DAS PROGRAMM OPENFIL, CLOSEFIL

ÜBERGEORDNETES PROGRAMM : EMOS2

LÄNGE DES PROGRAMMS : 125 Zeilen
 747 Bytes

STAND DER DOKUMENTATION : 15. März 1986

KURZBESCHREIBUNG:

Öffnen und Schließen von Files der Datenbibliothek

PARAMETERLISTE :

OPENFIL IKAR, CTYPX
CLOSEFIL CTYPX

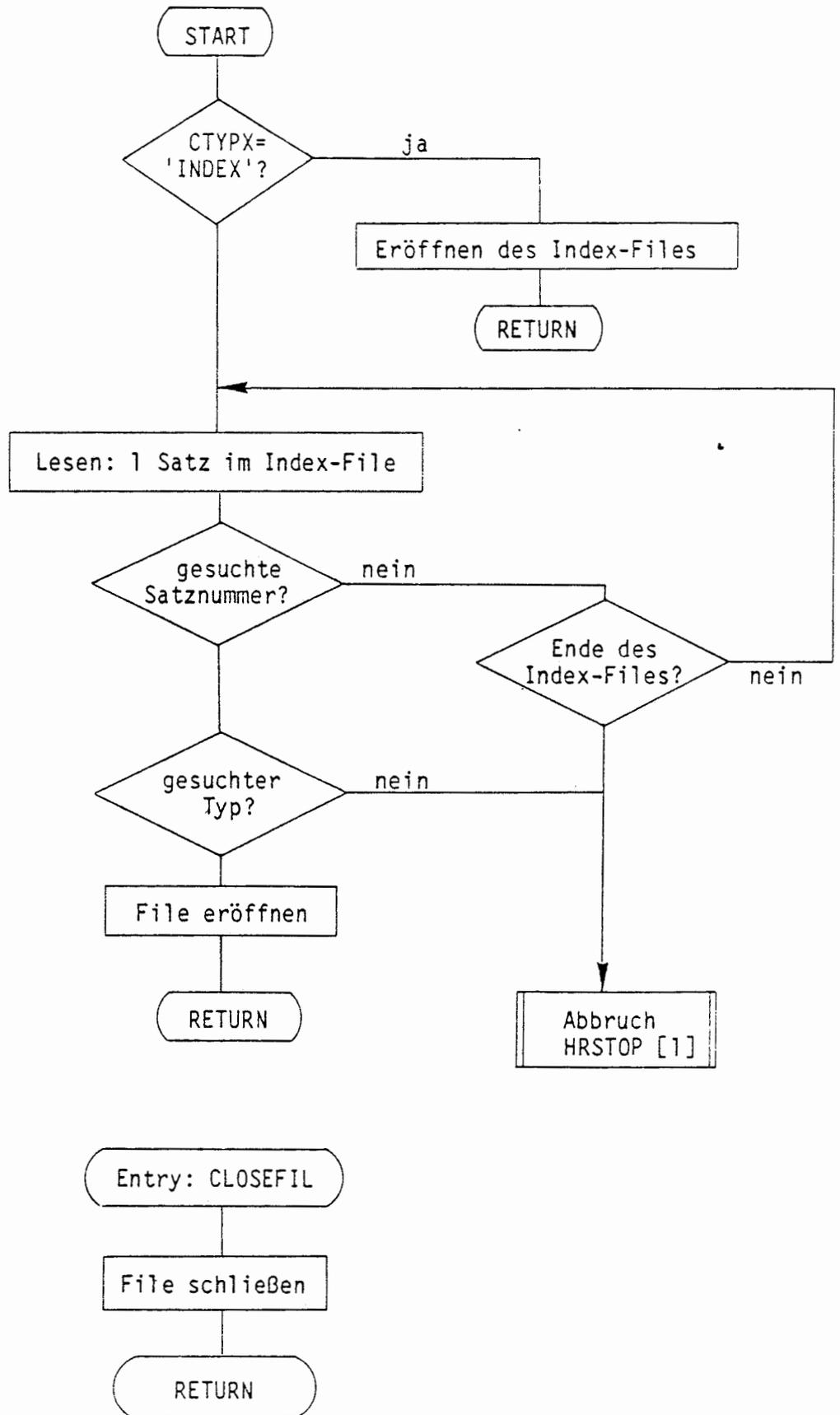
EINGANGSPARAMETER :

OPENFIL IKAR : Satznummer des Datenfiles
 CTYPX : Typ des Datenfiles
CLOSEFIL CTYPX : Typ des Datenfiles

UNTERPROGRAMMAUFRUFE:

[1] HRSTOP ('OPENFIL')

ABLAUF DES PROGRAMMS OPENFIL/ENTRY CLOSEFIL



A 1.3 HILFSROUTINE FÜR MODULE (HRANF)

Zur Durchführung von Hilfstätigkeiten beim formalen Ablauf eines Moduls existieren verschiedene Entries, für die zunächst jeweils die bereits verbrauchte Rechenzeit ermittelt und dann zusammen mit einer ENTRY-spezifischen Meldung ausgedruckt wird. Bei einigen Entries werden dann noch zusätzliche Aufgaben bearbeitet:

HRANF : - Ausgabe der Anfangsmeldung eines Moduls
- Zuweisung eines leeren Textwortes (BLANK)
- Lesen und Schreiben der Kommentarzeile

HREND : - Ausgabe der Schlußmeldung eines Moduls

HRSTOP : - Ausgabe eines Programmnamens, in dem ein fataler Fehler aufgetreten ist
- Programmabbruch

HRDRUBC: - Ausgabe einer Meldung

ÜBERSICHT ÜBER DAS PROGRAMM HRANF, HREND, HRSTOP, HRDRUBC

ÜBERGEORDNETES PROGRAMM : EMOS2

LÄNGE DES PROGRAMMS : 75 Zeilen
484 Bytes

STAND DER DOKUMENTATION : 15. März 1986

KURZBESCHREIBUNG:

Durchführung von Hilfstätigkeiten beim formalen Ablauf eines Moduls

PARAMETERLISTE:

CTEXT

EINGANGSPARAMETER:

CTEXT : Text zur Beschreibung des Anlasses der Hilfstätigkeit

LESENDER ZUGRIFF AUF COMMON-BLÖCKE:

HRANF : /WTEXT/: CTEXTI
HREND : / /: ZEIT
HRSTOP : / /: ZEIT
HRDRUBC : / /: ZEIT

SCHREIBENDER ZUGRIFF AUF COMMON-BLÖCKE:

HRANF : /WTEXT/: CBLANK, CTEXTI
HREND : / /: ZEIT
HRSTOP : / /: ZEIT
HRDRUBC : / /: ZEIT

A 1.4 HILFSFUNKTIONEN BEI DER AUSDRUCKSTEUERUNG (NOUTBAR)

Bei der Ausgabe von barrierenspezifischen und nuklidspezifischen Daten während der Dateneingabe (DALEKO) und der zeitdiskreten Rechnung (STEBA) ist jeweils zu entscheiden, ob ein betrachtetes Nuklid oder eine betrachtete Barriere in einer Auswahl enthalten ist.

Für eine vorgegebene Barriere entscheidet die FUNCTION NOUTBAR, ob eine Ausgabe für diese Barriere aufgrund der eingegebenen Steuerparameter gewünscht wird. Für ein vorgegebenes Nuklid entscheidet die FUNCTION NOUTNUK ob eine Ausgabe für dieses Nuklid gewünscht wird. Wenn eine Ausgabe erfolgen soll, dann ist der jeweilige Funktionswert nach dem Aufruf gleich eins.

Zur Berechnung von Massensummen für ausgewählte Nuklide muß bei der Vorgabe einer Kombination von Nuklidnamen und Atomgewicht die Position dieses Nuklid in der Liste der nuklidspezifischen Daten ermittelt werden, damit ein Zugriff auf die nuklidspezifischen Daten erfolgen kann. Dies wird durch die FUNCTION NUNUK geleistet. Falls das vorgegebene Nuklid nicht gefunden wird, erfolgt ein Programmabbruch.

ÜBERSICHT ÜBER DAS PROGRAMM NOUTBAR ,NOUTNUK, NUNUK

ÜBERGEORDNETES PROGRAMM :	EMOS2
LÄNGE DES PROGRAMMS :	187 Zeilen 349 Bytes
STAND DER DOKUMENTATION :	15. März 1986

KURZBESCHREIBUNG:

Die FUNCTION NOUTBAR bzw. NOUTNUK entscheidet durch ihren Funktionswert, ob eine Ausgabe für eine vorgegebene Barriere bzw. ein vorgegebenes Nuklid aufgrund entsprechender Auswahlen in den Steuerparametern gewünscht wird.

Die FUNCTION NUNUK ermittelt für eine vorgegebene Kombination von Nuklidname und Atomgewicht die Position dieses Nuklids in der Nuklidmatrix.

PARAMETERLISTE:

NOUTBAR : KBY
NOUTNUK : KNY
NUNUK : CN, N

EINGANGSPARAMETER:

NOUTBAR : KBY - Barrierenname
NOUTNUK : KNY - Nuklidnummer
NUNUK : CN - Nuklidname
N - Atomgewicht

AUSGANGSPARAMETER:

NOUTBAR : Schalter zum Ausdruck
NOUTNUK : Schalter zum Ausdruck
NUNUK : Nuklidnummer

LESENDER ZUGRIFF AUF COMMON-BLÖCKE:

NOUTBAR /RZ/ : IAU(1)
/WCBY/ : CBY(1,KBY)
/RC/ : CBAU

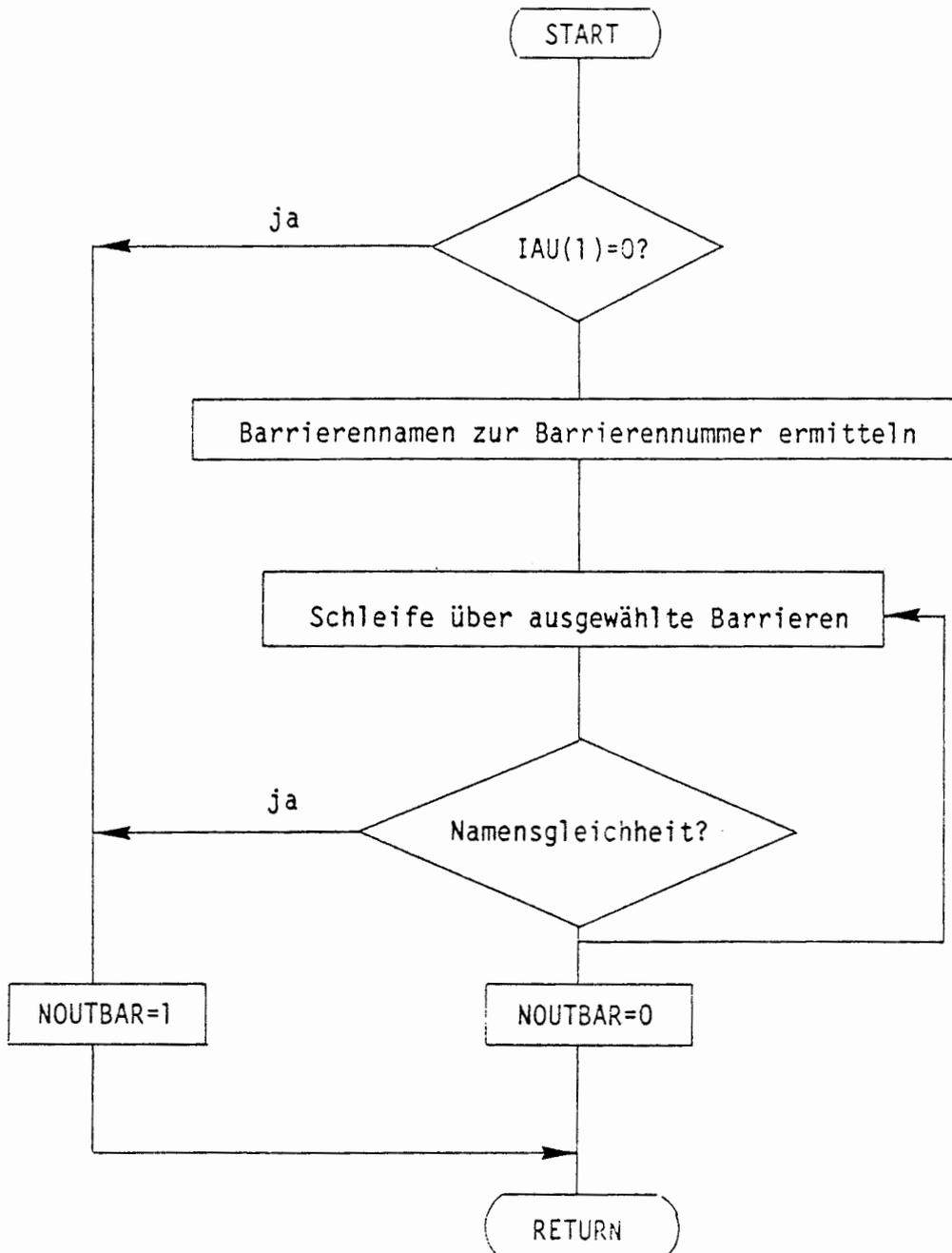
NOUTNUK /RZ/ : IAU(2) JAU
/WCNY/ : CNY(1,KNY)
/RC/ : CNAU
/WZNY/ : INY(1,KNY)

NUNUK /WZNY/ : NNY, INY(1,KNY)
/WCNY/ : CNY(1,KNY)

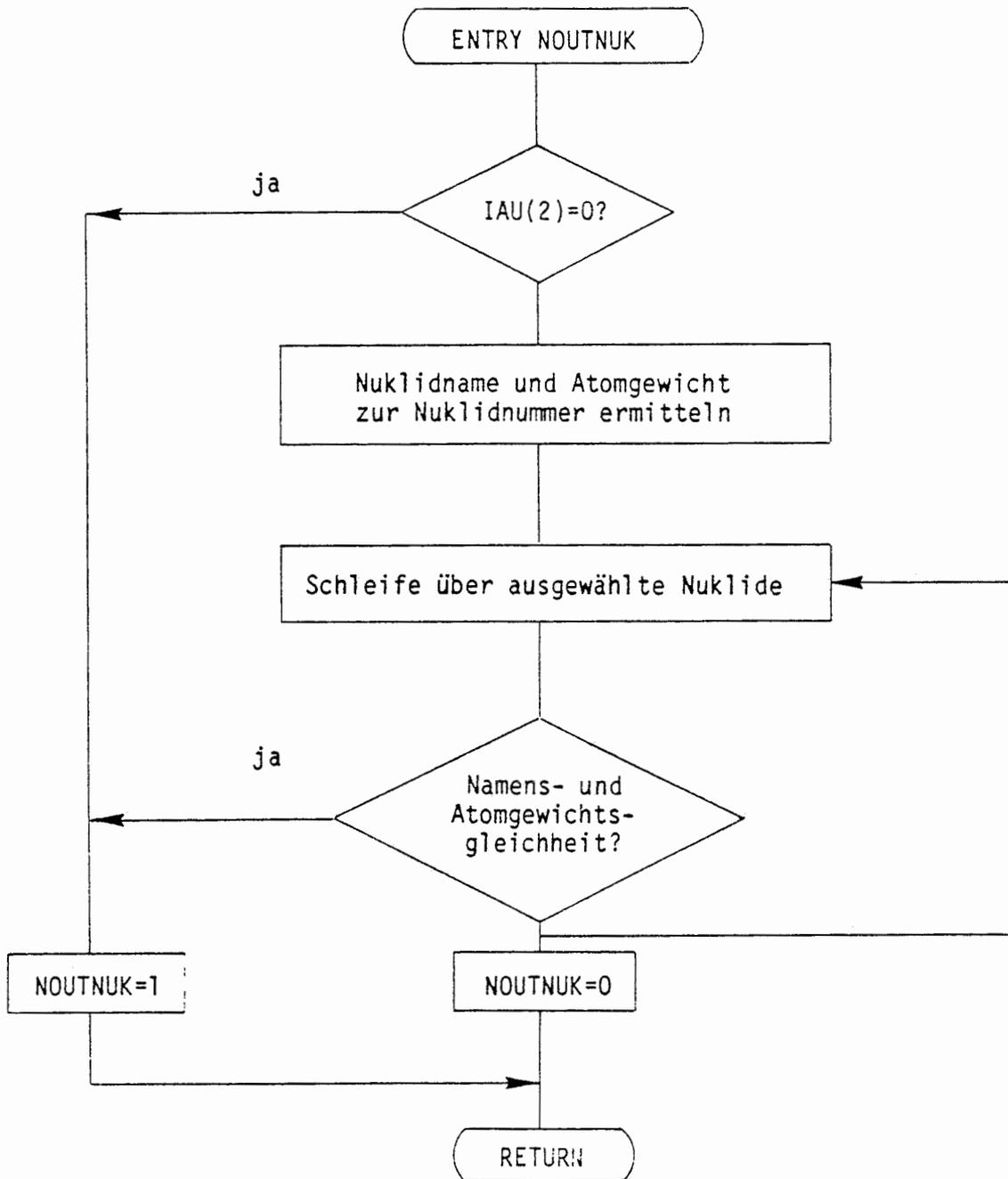
UNTERPROGRAMMAUFRUFE:

[1] HRSTOP('NUNUK')

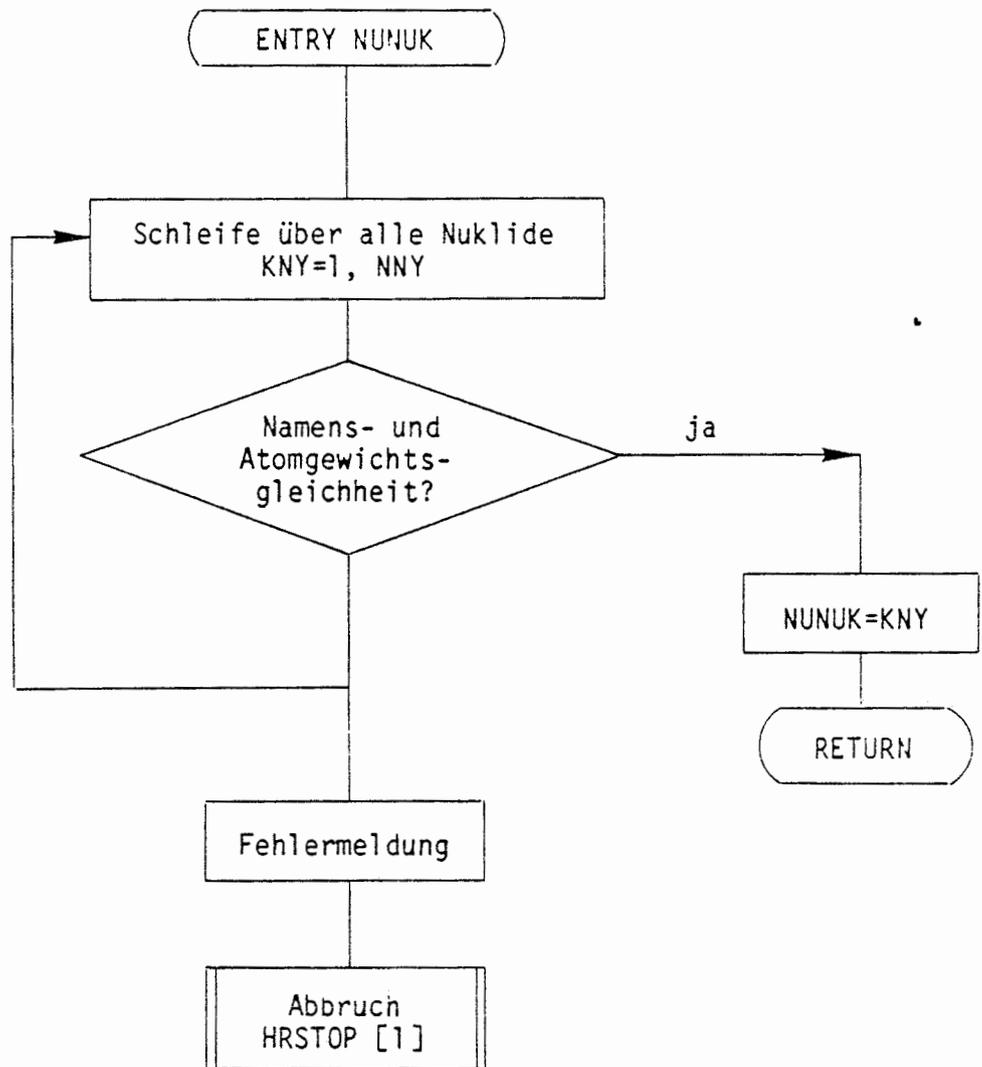
ABLAUF DES PROGRAMMS NOUTBAR



ABLAUF DES PROGRAMMS NOUTBAR/ENTRY NOUTNUK



ABLAUF DES PROGRAMMS NOUTBAR/ENTRY NUNUK



A 2 RADIONUKLIDFREISETZUNG AUS EINEM GRUBENGEBÄUDE

Im folgenden werden nach einer Erläuterung der Programmkonzeption zur Durchführung von Freisetzungsberechnungen sämtliche Unterprogramme beschrieben, die zur Berechnung der Radionuklidfreisetzungen aus einem Grubengebäude erforderlich sind. Dabei werden nur solche Unterprogramme behandelt, die zur Berechnung der Freisetzung aus einem Grubengebäude in einer Erzformation gebraucht werden.

A 2.1 PROGRAMMKONZEPT FÜR FREISETZUNGSRECHNUNGEN

Zur Berechnung der Radionuklidfreisetzung aus einem Grubengebäude müssen folgende Hauptprozesse behandelt werden:

- Mobilisierung von Radionukliden aus Abfallgebänden und
- Ausbreitung in und Transport aus Teilbereichen des Grubengebäudes

Der Transport aus demjenigen Teilbereich des Grubengebäudes, der mit dem Deckgebirge in Verbindung steht, liefert dann die Freisetzung aus dem Grubengebäude ins Deckgebirge.

Unter Teilbereichen des Grubengebäudes werden hier solche Bereiche verstanden, die eine weitgehend abgeschlossene Einheit darstellen. Solche Teilbereiche sind z.B. verschlossene Bohrlöcher, verschlossene Kammern sowie verschlossene oder abgedämmte Streckenabschnitte.

Wegen der großen Anzahl von Abfallgebänden und Teilbereichen in einem Grubengebäude sind die oben genannten Hauptprozesse relativ häufig zu betrachten. Ein einzelner Hauptprozess im Gesamtsystem des Grubengebäudes wird im folgenden als Barriere bezeichnet. Die Verknüpfung aller Barrieren zueinander wird in einer Barrierenstruktur dargestellt.

Da es sich bei verschiedenen Hauptprozessen oftmals um identische Vorgänge handelt, wird eine Berechnung nur für repräsentative Barrieren durchgeführt. Solche sind zum Beispiel:

- Mobilisierung aus einer HAW-Kokille in das Lösungsvolumen eines HAW-Bohrloches
- Transport aus einem Bohrloch in die darüberliegende Beschickungsstrecke

Die entsprechenden Auswirkungen auf benachbarte Barrieren erhält man, indem die jeweiligen Übergangsströme mit der relativen Häufigkeit gewichtet werden. Beispiele für solche relativen Häufigkeiten sind:

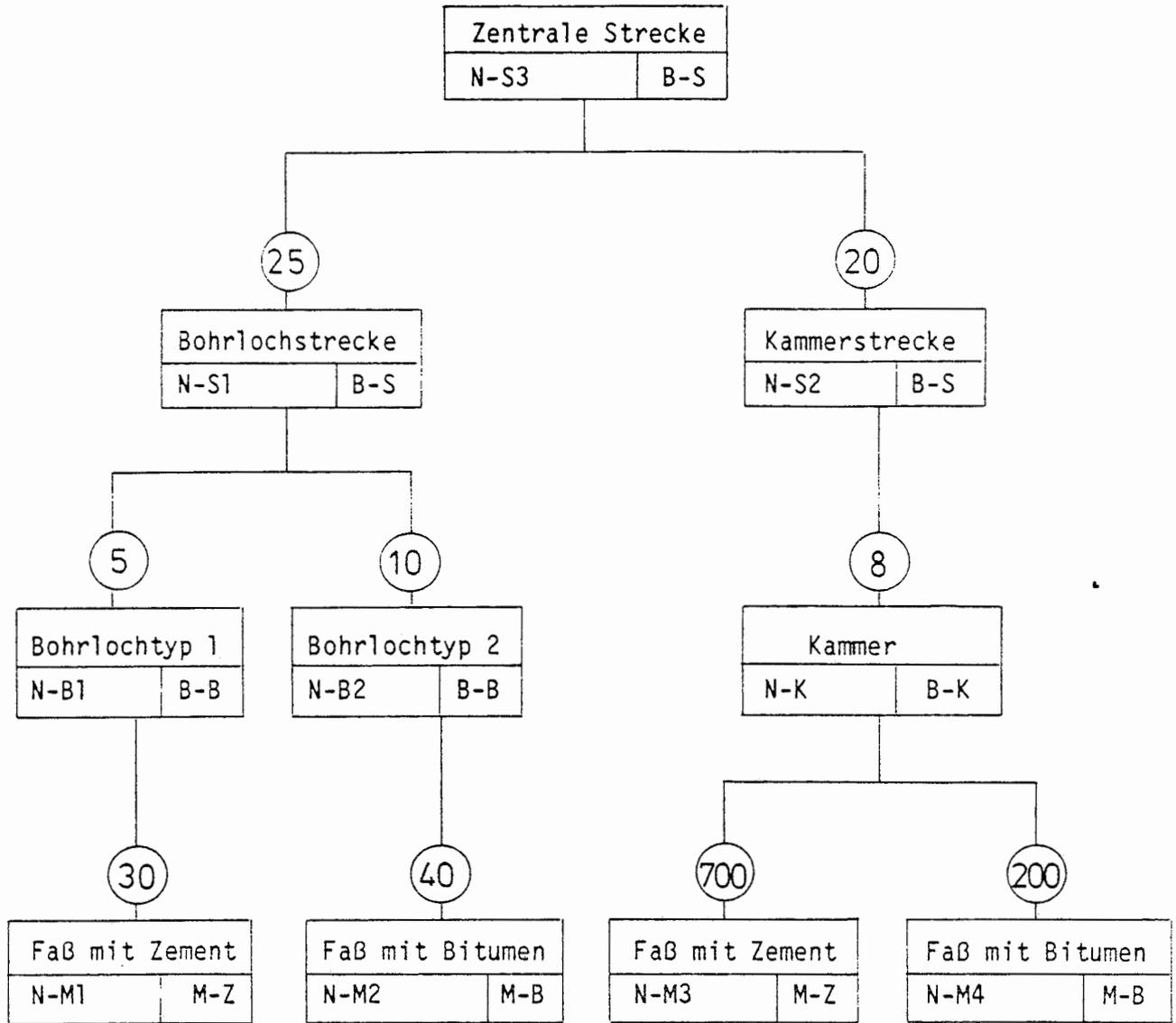
- Anzahl der Kokillen in einem Bohrloch
- Anzahl von Bohrlöchern in einer Beschickungsstrecke

Zur Beschreibung der Vorgänge in einer Barriere werden Unterprogramme mit einer standardisierten Parameterliste verwendet. Solche Unterprogramme werden als Barrierenmodelle bezeichnet; im speziellen Fall der Mobilisierung werden diese auch als Mobilisierungsmodelle bezeichnet. Wegen der standardisierten Parameterliste können die Barrierenmodelle in nahezu beliebiger Weise miteinander verknüpft werden. Bei fortschreitender Entwicklung lassen sich die Barrierenmodelle ebenfalls leicht gegen neuere Versionen austauschen.

Oftmals handelt es sich bei den Vorgängen in verschiedenen Barrieren um identische Hauptprozesse, so daß nicht für jede Barriere ein eigenes Barrierenmodell erforderlich ist. Beispiele hierfür sind

- Mobilisierungsmodelle für Rollreifenfässer, die in verschiedenen Kammertypen oder
- Barrierenmodelle für Bohrlöcher, die in verschiedenen Typen von Einlagerungsfeldern

verwendet werden können. Als Zusammenfassung der bisherigen Ausführungen gibt Abbildung A 2-1 ein Beispiel für eine Barrierenstruktur mit Angabe von Namen für Barrieren und Barrierenmodellen sowie von relativen Häufigkeiten.



Erläuterung

- | | | |
|---|---|-------------------------|
| 1. Buchstabe | : | B: Barrierenmodell |
| | | M: Mobilisierungsmodell |
| | | N: Barrierenname |
| 2. Buchstabe für Barrierenmodelle | : | S: Strecke |
| | | B: Bohrloch |
| | | K: Kammer |
| 2. Buchstabe für Mobilisierungsmodelle: | | Z: Zement |
| | | B: Bitumen |

Abb. A 2-1: Fiktives Beispiel einer Barrierenstruktur

Der Hauptprozess in einem Teilbereich des Grubengebäudes setzt sich aus mehreren Einzeleffekten zusammen. Die Modellansätze für die Einzeleffekte sind nicht direkt in den Barrierenmodellen enthalten, sondern werden durch weitere Unterprogrammaufrufe zur Verfügung gestellt. Die Schnittstellen zu den Einzeleffekten sind nicht standardisiert.

Ein bestimmter Einzeleffekt ist oftmals in vielen Barrierenmodellen zu berücksichtigen. Dabei ist der Modellansatz dieses Effektes vielfach nicht vom Barrierenmodell abhängig. Beispiele hierfür sind

- Ausfällung von Radionukliden beim Erreichen von Löslichkeitsgrenzen
- Konvergenz von Hohlräumen des Grubengebäudes durch den Gebirgsdruck

Das Unterprogramm für einen bestimmten Einzeleffekt wird somit nur einmal erstellt und steht dann in den verschiedenen Barrierenmodellen zur Verfügung. Bei fortschreitender Entwicklung läßt sich das Unterprogramm leicht durch eine überarbeitete Version ersetzen.

Durch die Einführung von Barrieren sowie Barrierenmodellen und Einzeleffekten wird die hohe Flexibilität bei der Durchführung von Langzeitsicherheitsanalysen erreicht. Die Barrieren und Barrierenstrukturen werden über das Job-Input-File eines Rechenlaufs vorgegeben und lassen sich leicht einem veränderten Planungsstand für das Endlager oder einem anderen Endlagerkonzept anpassen. Ebenso können die Einzeleffekte ohne Änderung der Programmstruktur beliebig modifiziert werden, um sie an den Stand von Forschung und Entwicklung anzupassen.

A 2.2 PROGRAMMSTRUKTUR (REPOS)

Die Struktur eines Rechenprogramms für das beschriebene Konzept ergibt sich aus den Teilaufgaben, die zur Durchführung einer Freisetzungsberechnung erforderlich sind. Wesentliche Teilaufgaben sind zunächst die Dateneingabe und die Ansteuerung der Barrierenmodelle für eine vorgegebene Barrierenstruktur sowie die Ausgabe der Rechenergebnisse. Da die Freisetzung in einer zeitdiskreten Vorgehensweise mit variabler Zeitschrittweite ermittelt wird, ergibt sich als weitere wichtige Teilaufgabe die Steuerung der Zeitschrittweite. Die wichtigsten Teilaufgaben sind im Zusammenhang mit dem prinzipiellen Ablauf einer Freisetzungsberechnung in Abbildung A 2-2 angegeben. Die sich daraus ergebende Grobstruktur des Freisetzungsprogramms ist in Abbildung A 2-3 dargestellt. Danach werden vom Programm REPOS fünf Unterprogramme in der angegebenen Reihenfolge aufgerufen. Da außer diesen Aufrufen kein ausführbarer Programmtext vorhanden ist und der Datentransfer zwischen den Unterprogrammen über COMMON-Blöcke geschieht, wird auf eine weitere Erläuterung des Programms REPOS verzichtet.

Die Programme DALEKO bzw. STEBA führen umfangreiche Arbeiten in Zusammenhang mit der Übernahme von Daten bzw. der Ausführung der zeitdiskreten Rechnung durch und sind daher weiter untergliedert. Die vollständigen Verknüpfungspläne sind in den Abbildungen A 2-4 und A 2-5 dargestellt. Weitere Erläuterungen folgen in den Kapiteln A 2.4 und A 2.5.

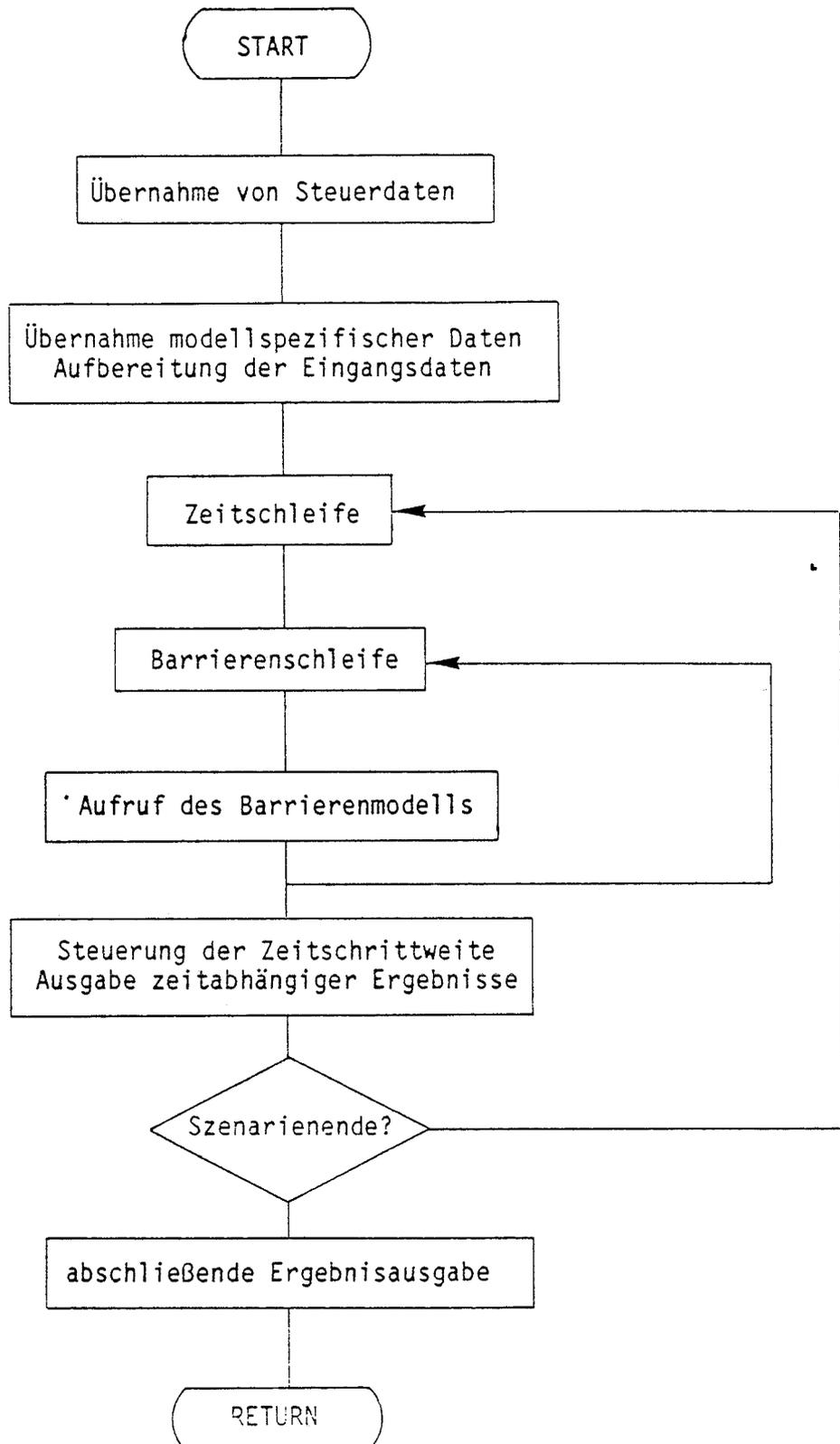
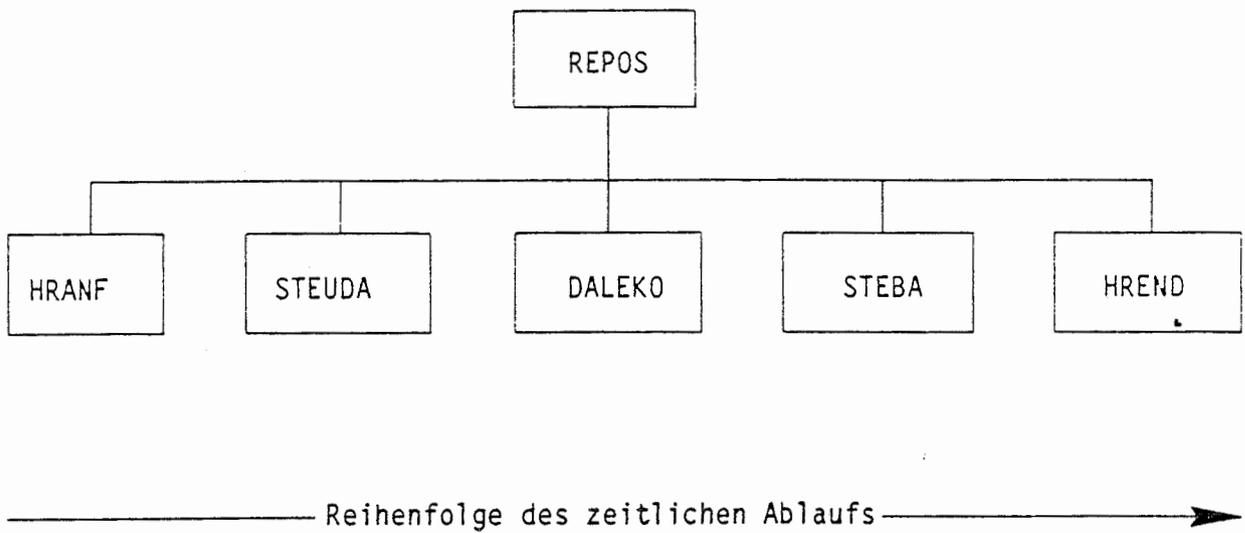


Abb. A 2-2: Grob Ablauf einer Freisetzungsberechnung



- HRANF, HREND : Starten und Beenden eines Moduls
- STEUDA : Übernahme von Steuerparametern
- DALEKO : Übernahme modellspezifischer Eingangsdaten
- STEBA : Berechnung der Freisetzung und Ergebnisausgabe

Abb. A 2-3: Grobstruktur eines Freisetzungsprogramms

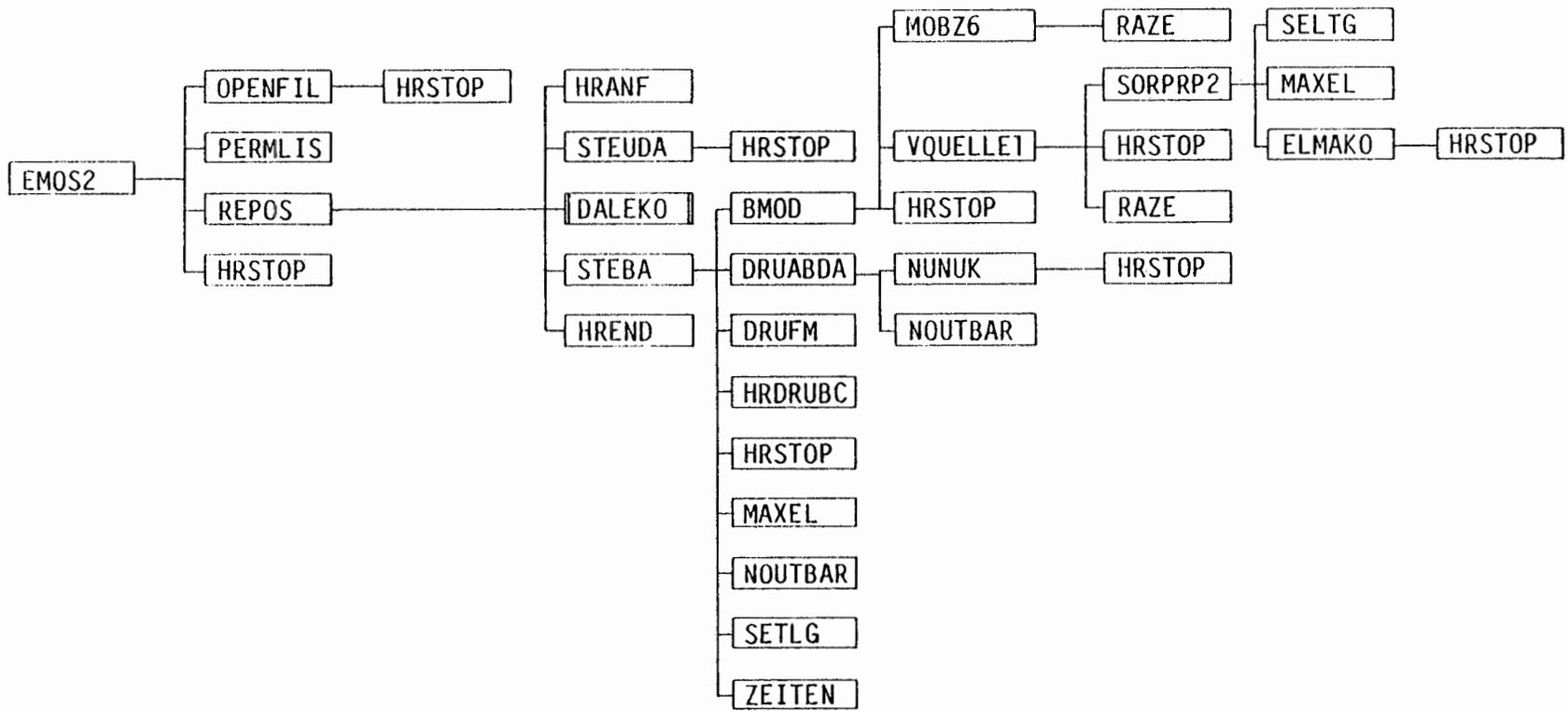


Abb. A 2-4: Bausteinverknüpfungsplan für EMOS

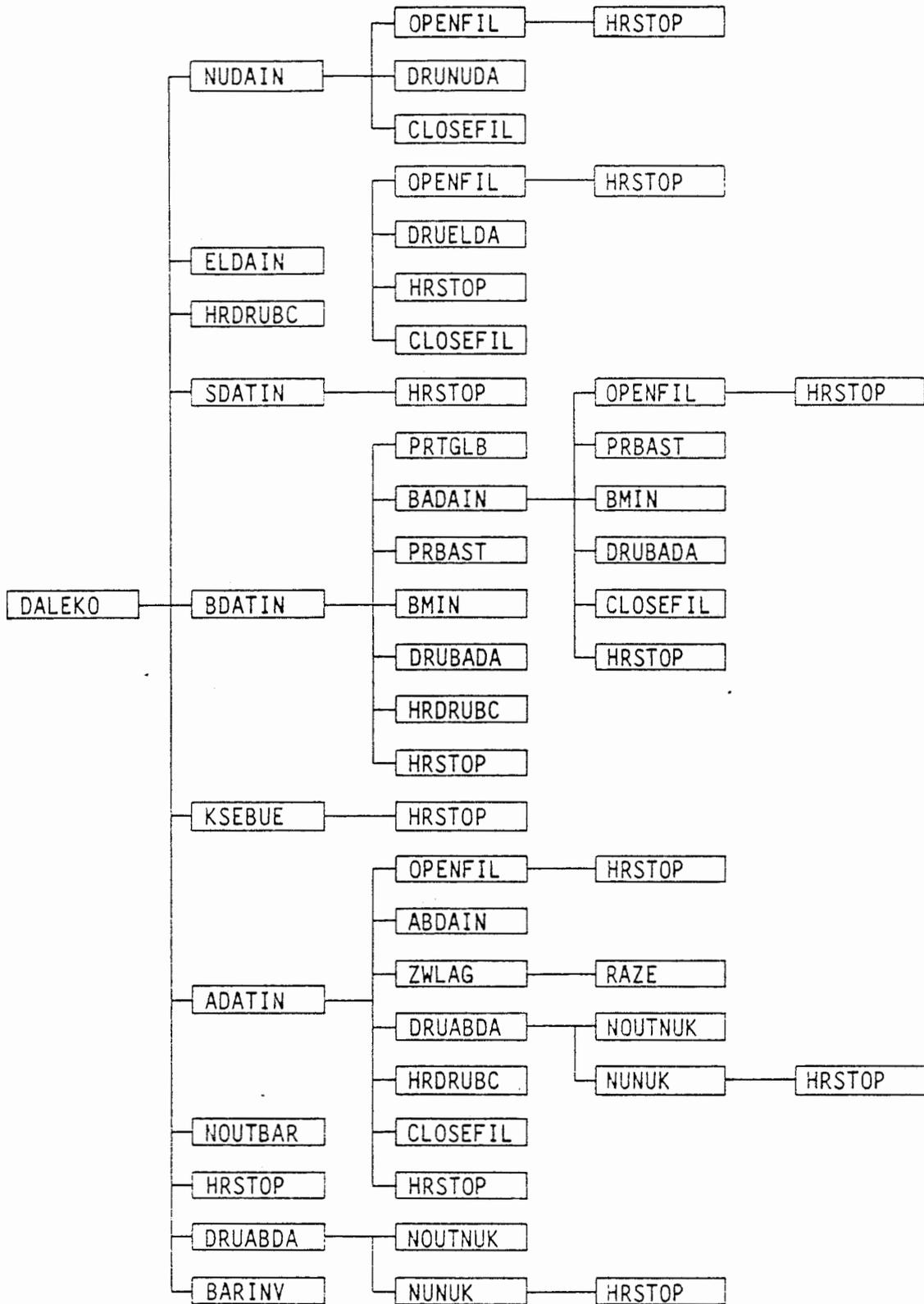


Abb. A 2-5: Bausteinverknüpfungsplan für DALEKO

A 2.3 EIN-/AUSGABE VON STEUERDATEN

Zur Steuerung des Ausdruckes von Eingabedaten und Rechenergebnissen werden eine Vielzahl von Steuerparameter eingelesen, überprüft und ausgedruckt. Daneben werden die, zusammen mit der Modulansteuerung eingelesenen Nummern der Datenfiles überprüft und ausgedruckt.

ÜBERSICHT ÜBER DAS PROGRAMM STEUDA

ÜBERGEORDNETES PROGRAMM : REPOS

LÄNGE DES PROGRAMMS : 346 Zeilen
4793 Bytes

STAND DER DOKUMENTATION : 15. März 1986

KURZBESCHREIBUNG:

Überprüfung und Ausgabe der Nummern von Datenfiles. Eingabe,
Überprüfung und Ausgabe von Steuerparameter zur Datenausgabe.

LESENDER ZUGRIFF AUF COMMON-BLÖCKE:

/ / : LFILE, IST
/RC/, /RZ/

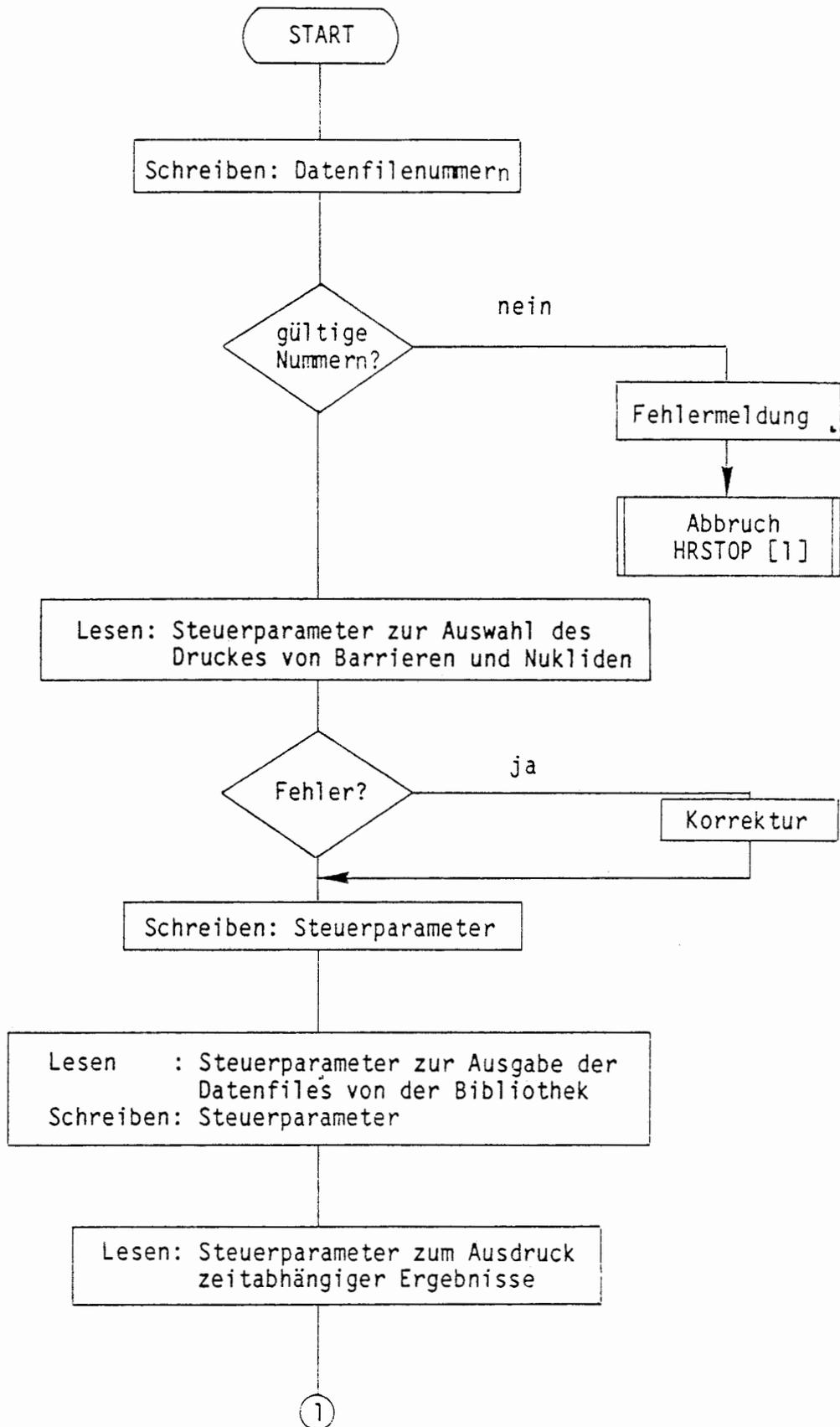
SCHREIBENDER ZUGRIFF AUF COMMON-BLÖCKE:

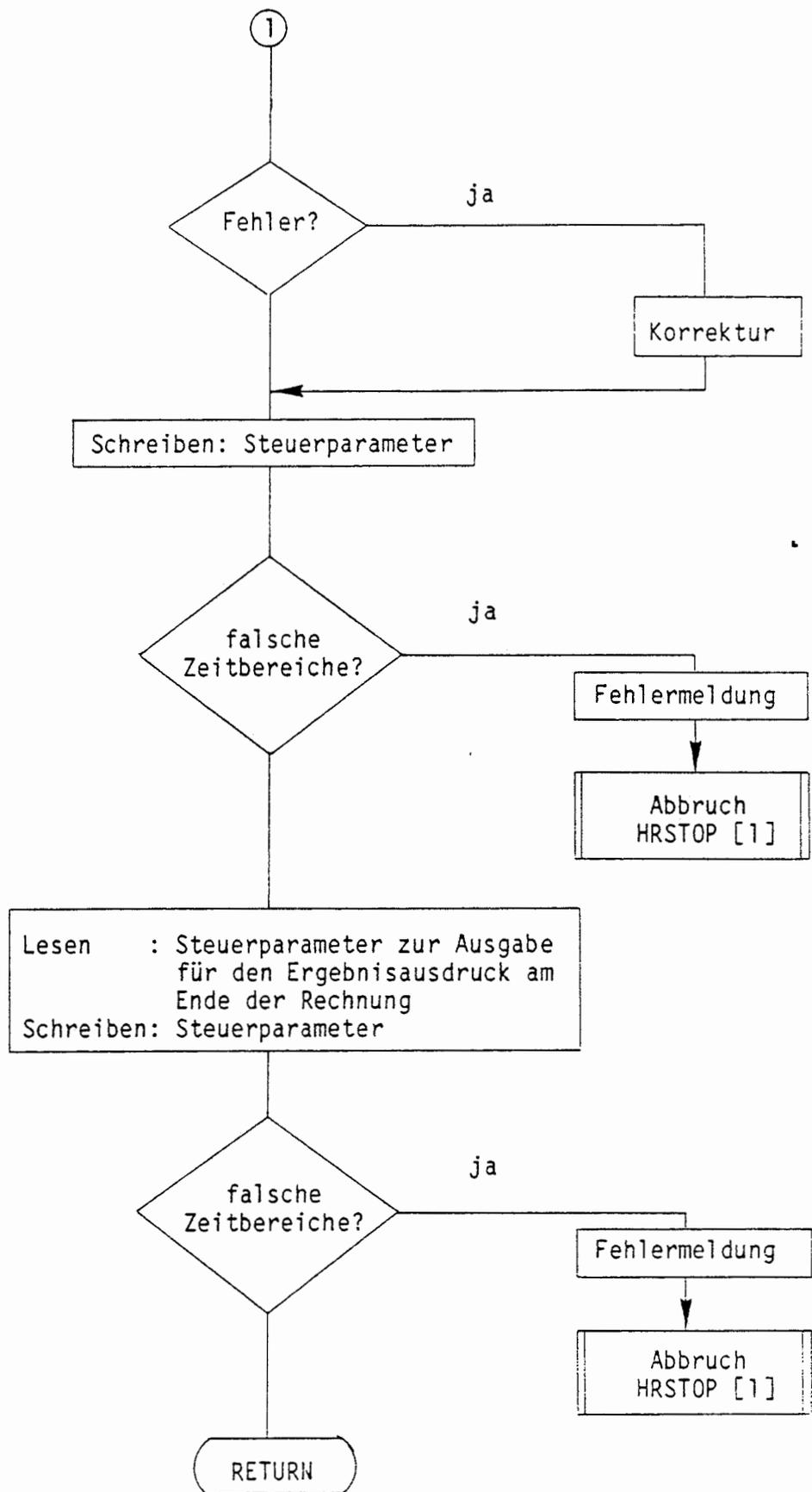
/RC/, /RZ/

UNTERPROGRAMMAUFRUFE:

[1] HRSTOP('STEUDA')

ABLAUF DES PROGRAMMS STEUDA





A 2.4 EIN-/AUSGABE MODELLSPEZIFISCHER EINGANGSDATEN (DALEKO)

DALEKO (Daten LEsen und KODieren) ist einer der größeren Programmteile von REPOS. Zur Vorbereitung der zeitdiskreten Rechnung werden alle modellspezifischen Eingangsdaten übernommen und auf Wunsch zur Kontrolle ausgedruckt. Die Übernahme der Daten erfolgt sowohl vom Job-Input-File als auch von der Datenbibliothek.

Die übernommenen Daten werden kontrolliert und aufbereitet und danach in einigen Hauptspeicherbereichen abgelegt, die dann für die zeitdiskrete Rechnung zur Verfügung stehen. Die übernommenen Datenarten und die daraus resultierenden Hauptspeicherbereiche sind wie folgt:

- Steuerparameter : /YZ/, /RZ/
- nuklidspezifische Daten : /WCNY/, /WZNY/
- elementspezifische Daten : /WCEY/, /WZEY/
- Strukturdaten : /WZSY/
- Barrierendaten : /WCBY/, /WZBY/, /G/
- abfallspezifische Daten : /WZAY/

Nach dem Einlesen der Parameter für die Steuerung der Zeitschrittweite vom Job-Input-File werden nuklid-, element- und ggf. barrierenspezifische Daten im wesentlichen von der Datenbibliothek übernommen. Nach Eingabe der Verknüpfungen für die Barrierenstruktur wird diese kodiert (Umsetzung der Barrierennamen in Nummern) und die Vollständigkeit der übergebenen Barrieren überprüft.

Zur Versorgung der entsprechenden Barrieren mit Radionuklidinventaren werden die abfallspezifischen Daten von der Datenbibliothek übernommen. Deren Vollständigkeit wird im Hinblick auf die Barrierendaten überprüft, so daß die Nuklidinventare für das gesamte Barrierensystem in kg und Bq bezogen auf den Beginn der Nachbetriebsphase des Endlagers auf Anforderung berechnet werden können.

Zur Organisation des Datentransfers zwischen den Barrierenmodellen bei der zeitdiskreten Rechnung werden sogenannte Übergangsmatrizen (RU1Y, RU2Y, RU3Y) eingerichtet, die die aktuellen Ausgangsströme einer jeden Barriere aufnehmen. Zur Berechnung kumulierter Werte aus den Ausgangsströmen insbesondere im Hinblick auf Nuklidfreisetzungsmengen werden entsprechende Speicherbereiche (RK1Y, RK2Y) initialisiert.

ÜBERSICHT ÜBER DAS PROGRAMM DALEKO

ÜBERGEORDNETES PROGRAMM : EMOS

LÄNGE DES PROGRAMMS : 460 Zeilen
 2704 Bytes

STAND DER DOKUMENTATION : 15. März 1986

KURZBESCHREIBUNG:

Übernahme aller modellspezifischen Eingangsdaten und Aufbereitung für eine zeitdiskrete Rechnung.

LESENDER ZUGRIFF AUF COMMON-BLÖCKE:

/ /, /RZ/, /RC/

SCHREIBENDER ZUGRIFF AUF COMMON-BLÖCKE:

/YZ/, /WZUY/, /WZKY/

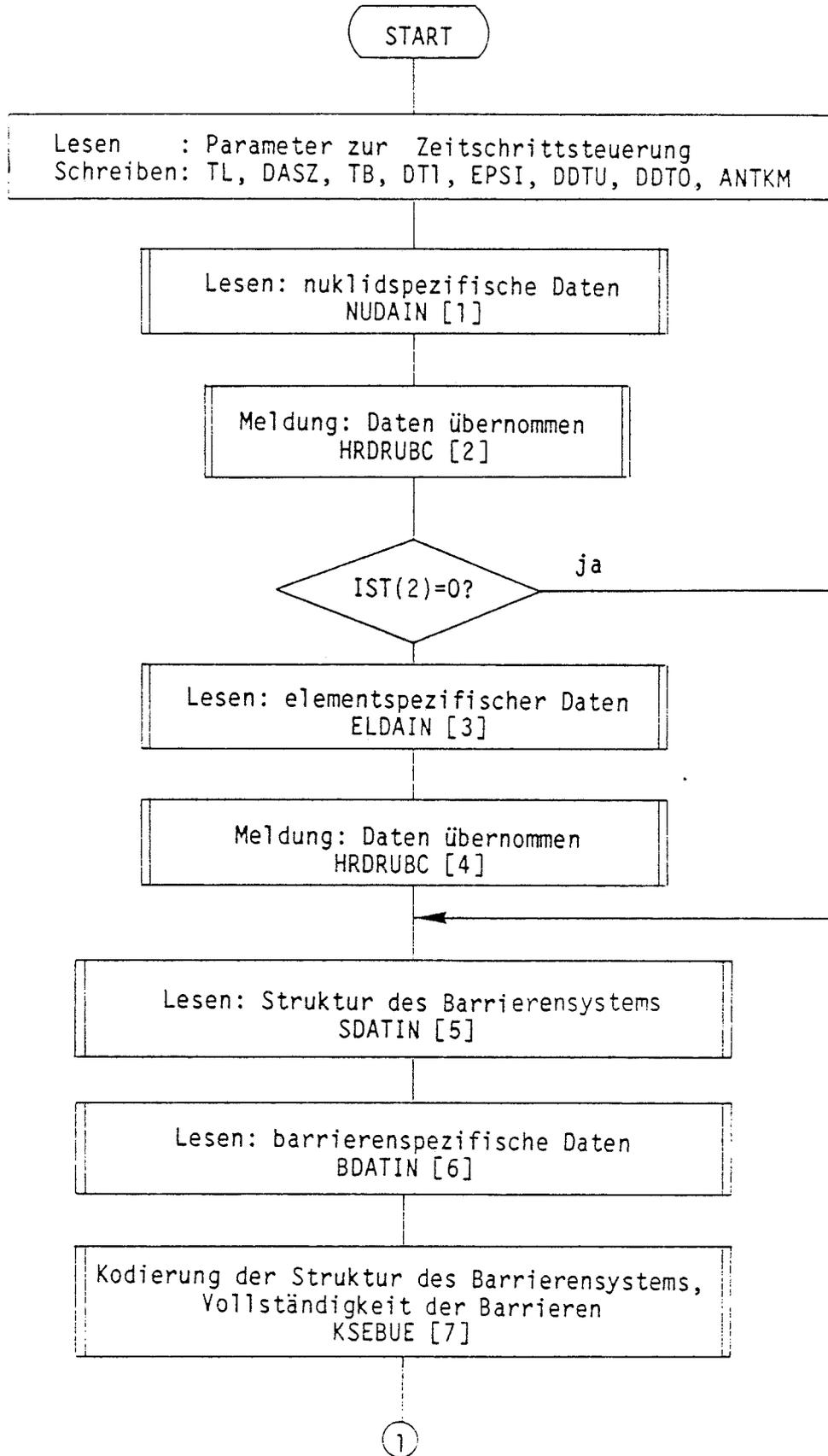
ZUGRIFF BEIM UNTERPROGRAMMAUFRUF:

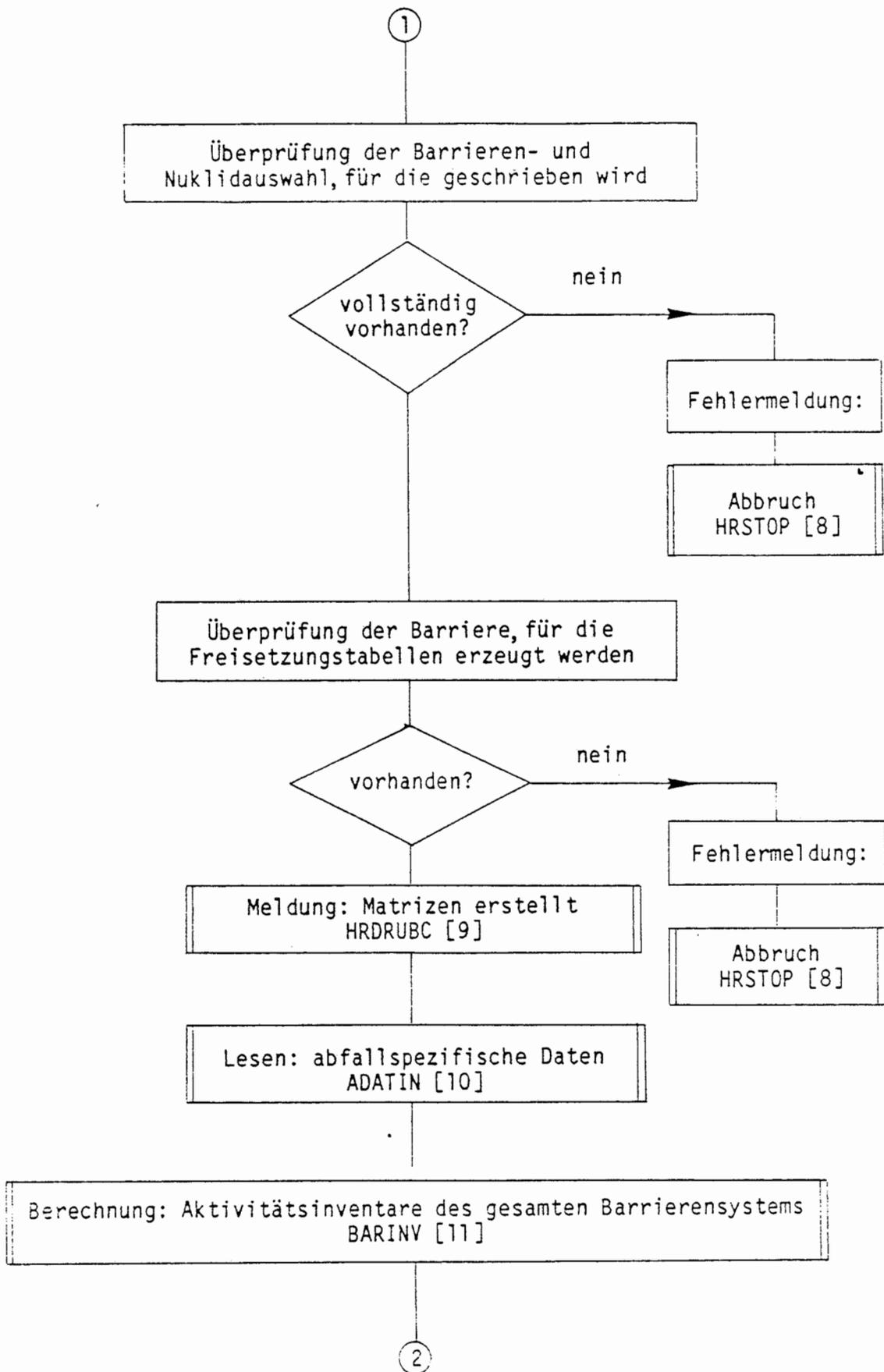
/WZNY/, /WCBY/, /WZAY/, /WZIY/, / /

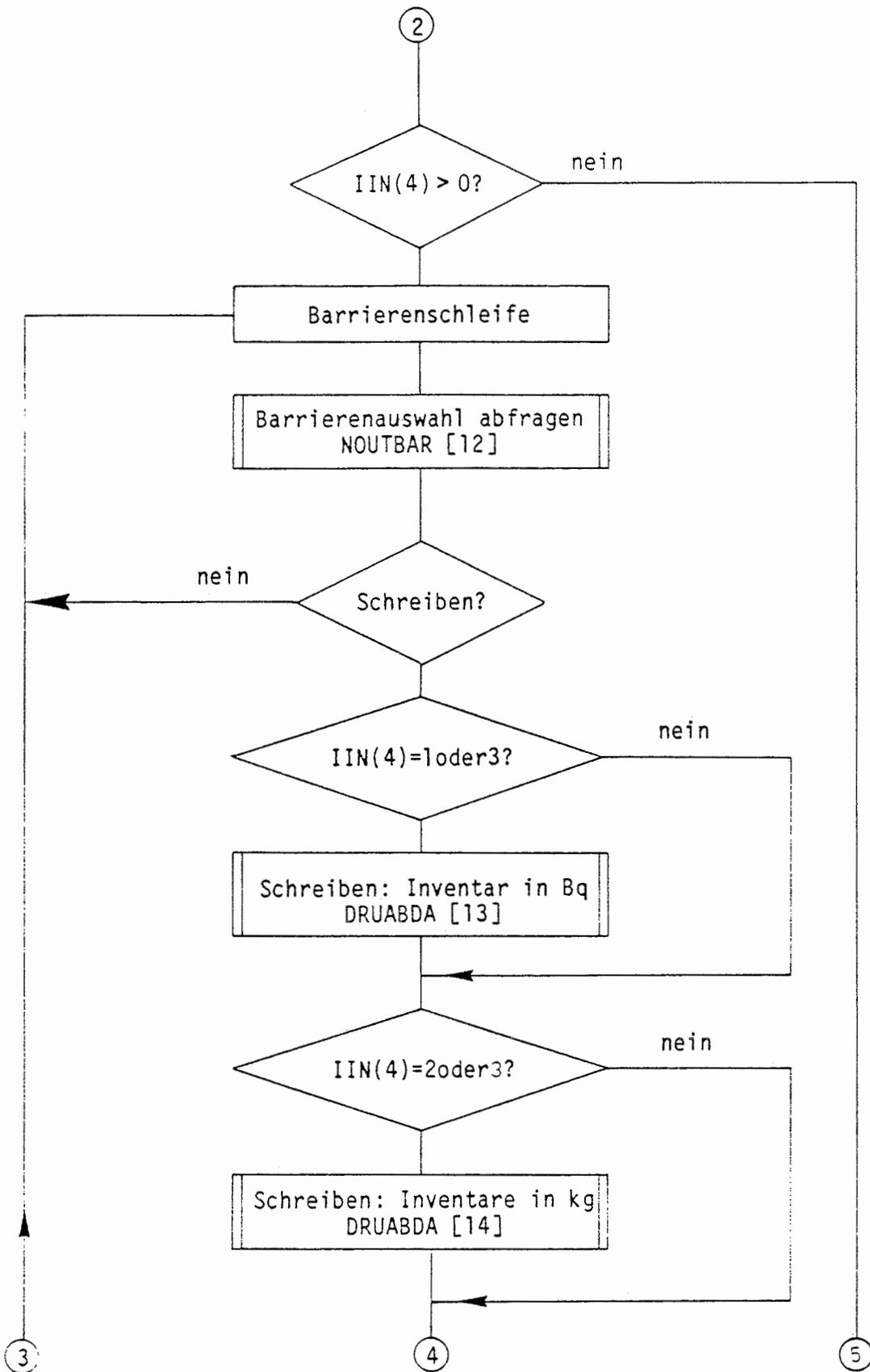
UNTERPROGRAMMAUFRUFE:

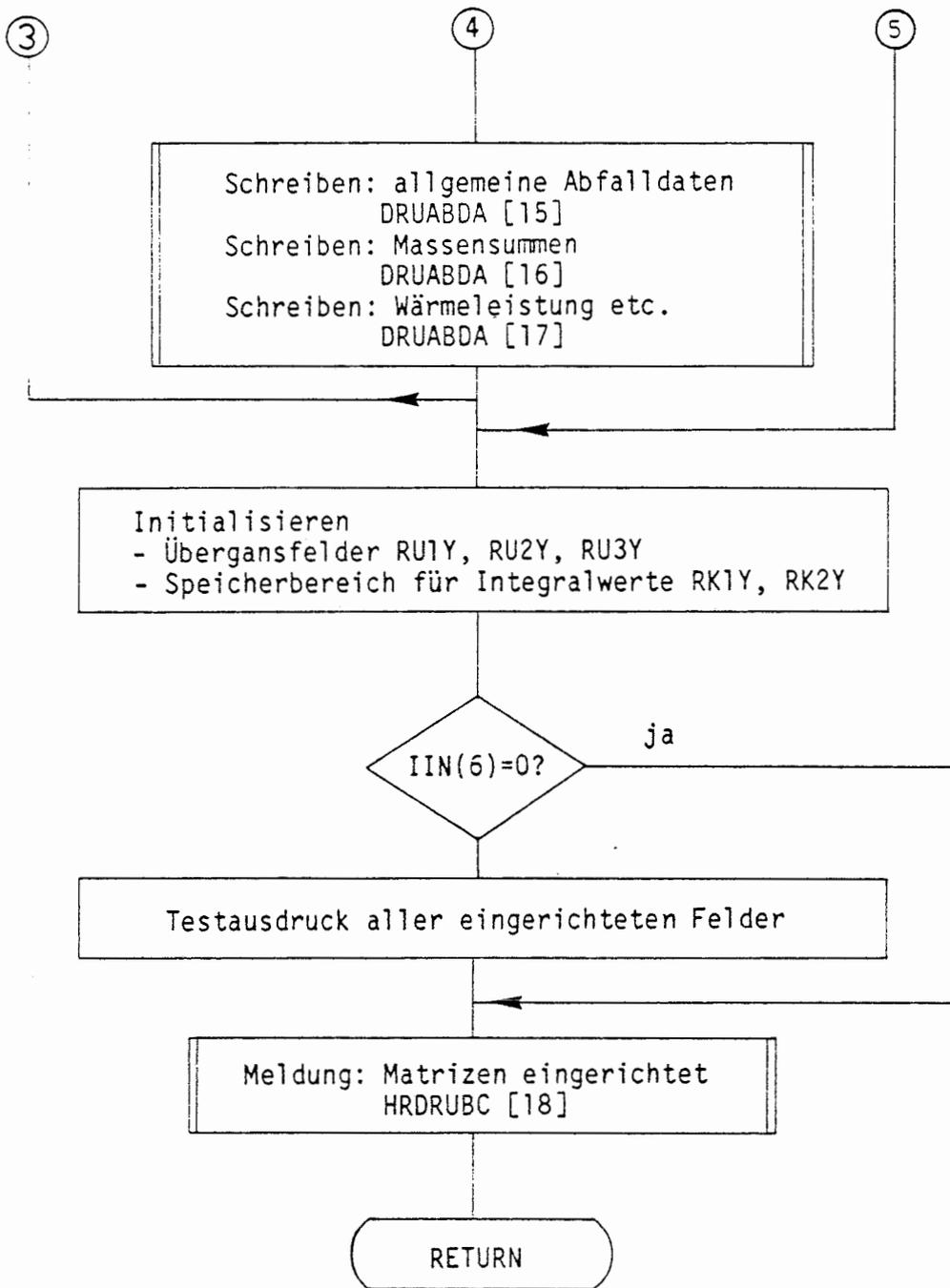
- [1] NUDAIN(IST(1))
- [2] HRDRUBC('NUKLIDSPEZIFISCHE DATEN ÜBERNOMMEN')
- [3] ELDAIN(IST(2))
- [4] HRDRUBC('ELEMENTSPEZIFISCHE DATEN ÜBERNOMMEN')
- [5] SDATIN
- [6] BDATIN
- [7] KSEBUE
- [8] HRSTOP('DALEKO')
- [9] HRDRUBC('STRUKTURMATRIX UND BARRIERENMATRIX EINGERICHTET')
- [10] ADATIN
- [11] BARINV
- [12] Funktion NOUTBAR(KBY)
- [13] DRUABDA(RIY(1, KBY), NNY, DUMMY, 1, CBY(1, KBY), -1)
- [14] DRUABDA(RIY(1, KBY), NNY, DUMMY, 1, CBY(1, KBY), -2)
- [15] DRUABDA(RA2Y(1, KBY), LRA2Y, DUMMY, 1, CBLANK, 6)
- [16] DRUABDA(RIY(1, KBY), NNY, DUMMY, 1, CBLANK, 5)
- [17] DRUABDA(RIY(1, KBY), NNY, DUMMY, 1, CBLANK, 4)
- [18] HRDRUBC('ÜBERGANGSMATRIZEN EINGERICHTET')

ABLAUF DES PROGRAMMS DALEKO









A 2.4.1 NUKLIDSPEZIFISCHE DATEN

A 2.4.1.1 EINGABE NUKLIDSPEZIFISCHER DATEN (NUDAIN)

NUDAIN übernimmt nuklidspezifische Daten aus der Datenbibliothek. Gelesen werden die Anzahl der Spaltprodukte, die Anzahl der Nuklide in jeder der vier Zerfallsreihen und dann für jedes Nuklid jeweils:

Nuklidname : CNY(1, KNY)
Atomgewicht : INY(1, KNY)
Abstand des Nuklids zu einem
Tochternuklid in einer Zerfallsreihe : INY(2, KNY)
Strahlungstyp : INY(3, KNY)
Halbwertszeit : RNY(1, KNY)
Dosiskonversionsfaktor : RNY(3, KNY)
Wärmekonversionsfaktor : RNY(4, KNY)

In dem Programm werden die Zerfallsrate und der Massenkonzentrationsfaktor wie folgt berechnet:

Zerfallsrate : $\ln(2.0)/\text{Halbwertszeit}$

$RNY(2,KNY) = DLOG(2.0)/RNY(1,KNY)$

Massenkonzentrationsfaktor: $\frac{\text{Atomgewicht}}{\text{Loschmidtzahl}} \cdot \frac{\text{Halbwertszeit}}{\ln(2.0)} \cdot \frac{31.550.000 \text{ s}}{a} \cdot \frac{\text{kg}}{1000 \text{ g}}$

$RNY(5,KNY) = INY(1,KNY) \cdot RNY(1, KNY) \cdot 7.554E-20$

Falls $IIN(1) > 0$, ruft NUDAIN das Unterprogramm DRUNUDA auf, um die nuklidspezifischen Daten ins Output-File zu schreiben.

ÜBERSICHT ÜBER DAS PROGRAMM NUDAIN

ÜBERGEORDNETES PROGRAMM : DALEKO
LÄNGE DES PROGRAMMS : 126 Zeilen
366 Bytes
STAND DER DOKUMENTATION : 15. März 1986

KURZBESCHREIBUNG:

Übernahme der nuklidspezifischen Daten

PARAMETERLISTE:

KANAL

EINGANGSPARAMETER:

KANAL : Nummer des Datenfiles

LESENDER ZUGRIFF AUF COMMON-BLÖCKE:

/WZNY/
/RZ/ : IIN(1)

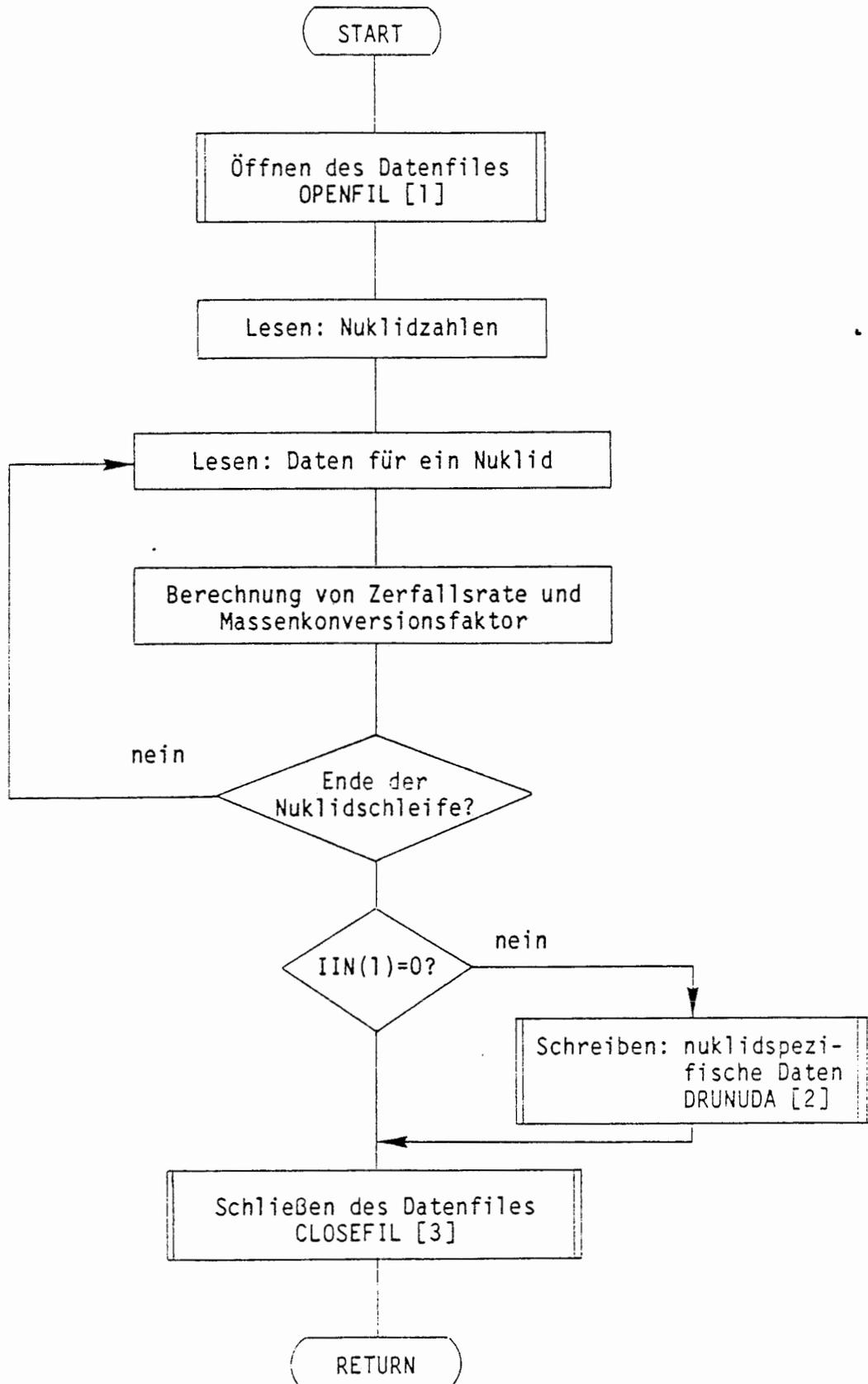
SCHREIBENDER ZUGRIFF AUF COMMON-BLÖCKE:

/WZNY/, /WCNY/

UNTERPROGRAMMAUFRUFE:

[1] OPENFIL (KANAL, 'NUDA')
[2] DRJNUDA
[3] CLOSEFIL ('NUDA')

ABLAUF DES PROGRAMMS NUDAIN



A 2.4.1.2 AUSGABE NUKLIDSPEZIFISCHER DATEN (DRUNUDA)

DRUNUDA schreibt nuklidspezifische Daten in das Output-File. In diesem Unterprogramm wird zuerst das Maximum der Anzahl der Nuklide in einer der fünf Nuklidgruppen (Spaltprodukte und vier Zerfallsreihen) ermittelt. Danach wird die Zahl der Nuklide in fünf Nuklidgruppen sowie die Gesamtnuklidzahl in das Output-File geschrieben.

Anschließend werden die Daten für alle Nuklide jeweils nach Datenart unterteilt geschrieben. Die sieben Unterteilungen beinhalten die folgenden Daten: Zerfallstyp, Strahlungstyp, Halbwertszeit, Zerfallsrate, Dosiskonversionsfaktoren, Wärmekonversionsfaktoren und Massenkonzentrationsfaktoren.

ÜBERSICHT ÜBER DAS PROGRAMM DRUNUDA

ÜBERGEORDNETES PROGRAMM :	DALEKO
LÄNGE DES PROGRAMMS :	139 Zeilen 648 Bytes
STAND DER DOKUMENTATION :	15. März 1986

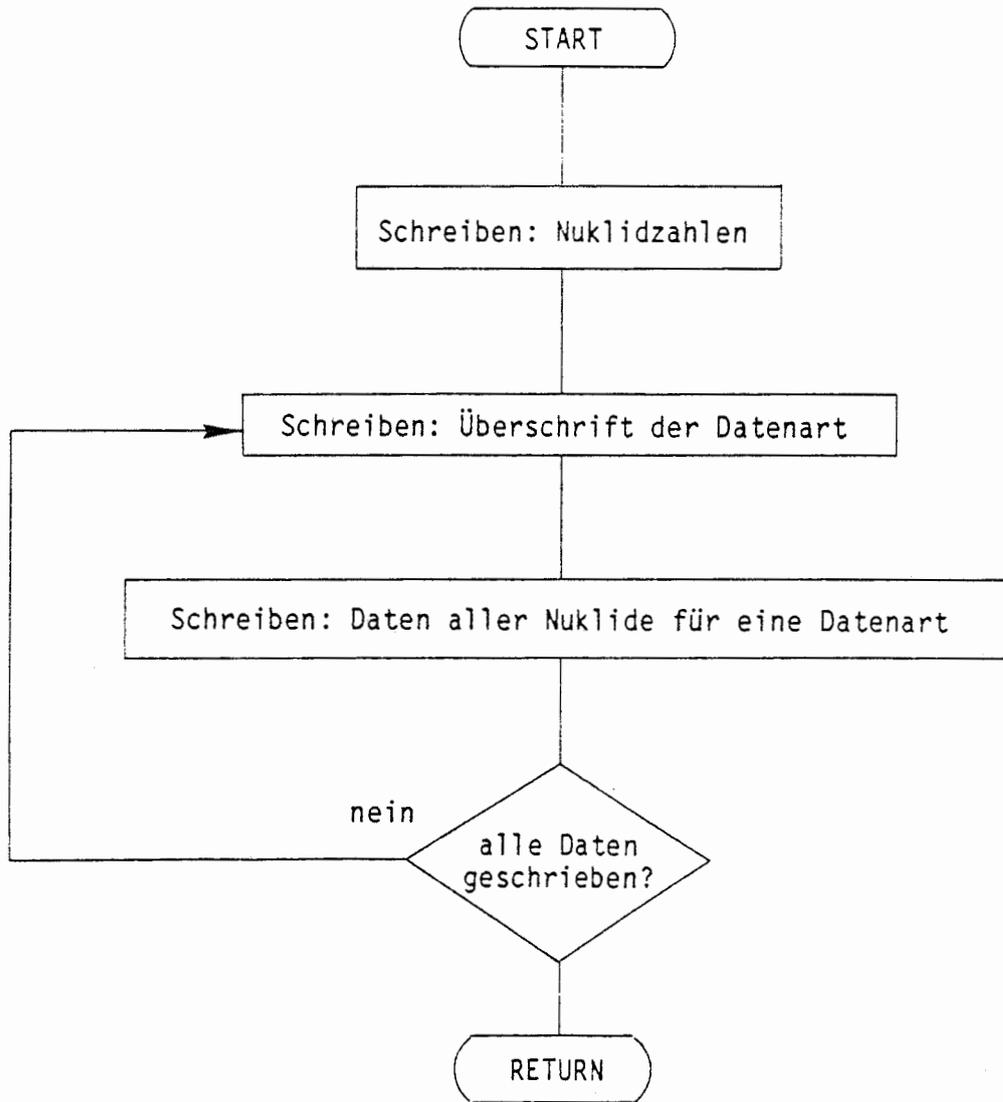
KURZBESCHREIBUNG:

Schreiben nuklidspezifischer Daten in das Output-File.

LESENDER ZUGRIFF AUF COMMON-BLÖCKE:

/WCNY/, /WZNY/

ABLAUF DES PROGRAMMS DRUNUDA



A 2.4.2 ELEMENTSPEZIFISCHE DATEN

A 2.4.2.1 EINGABE ELEMENTSPEZIFISCHER DATEN (ELDAIN)

ELDAIN liest die elementspezifischen Daten aus der Bibliothek. Die elementspezifischen Daten sind die chemischen Daten (z.B. Löslichkeitsgrenzen, KD-Werte) und die Mobilisierungsdauern der Elemente aus verschiedenen Gebinden (z.B. Zement, Kunststoff, Metall).

Gelesen werden die folgenden Daten

Anzahl der Elemente	:	NEY
Anzahl der Real*8-Größen		
pro Element	:	NREY
Bezeichnung des Datentyps:		CETEXT(KREY)
Elementname	:	CEY (1,KEY)
Elementspezifische Daten :		REY (KREY,KEY)

Eine bestimmte elementspezifische Datenart kann über die Zeilennummer in der Elementmatrix REY angesprochen werden.

Für jedes Nuklid wird die Spaltennummer des zugehörigen Elements in Elementmatrix ermittelt. Als Ergänzung des nuklidspezifischen Datensatzes wird diese Nummer im Vektor INY(4,KNY) abgespeichert. Die Vollständigkeit dieser Zuordnung wird überprüft. Bei Unvollständigkeit kommt es zum Programmabbruch.

Auf Anforderung (IIN(2)>0) werden die elementspezifischen Daten in das Output-File geschrieben.

ÜBERSICHT ÜBER DAS PROGRAMM ELDAIN

ÜBERGEORDNETES PROGRAMM : DALEKO
LÄNGE DES PROGRAMMS : 170 Zeilen
696 Bytes
STAND DER DOKUMENTATION : 15. März 1986

KURZBESCHREIBUNG:

Einlesen elementspezifischer Daten.

PARAMETERLISTE:

KANAL

EINGANGSPARAMETER:

KANAL : Nummer des Datenfiles

LESENDER ZUGRIFF AUF COMMON-BLÖCKE:

/WCNY/
/WCEY/ : CETEXT
/RZ/ : IIN(2)

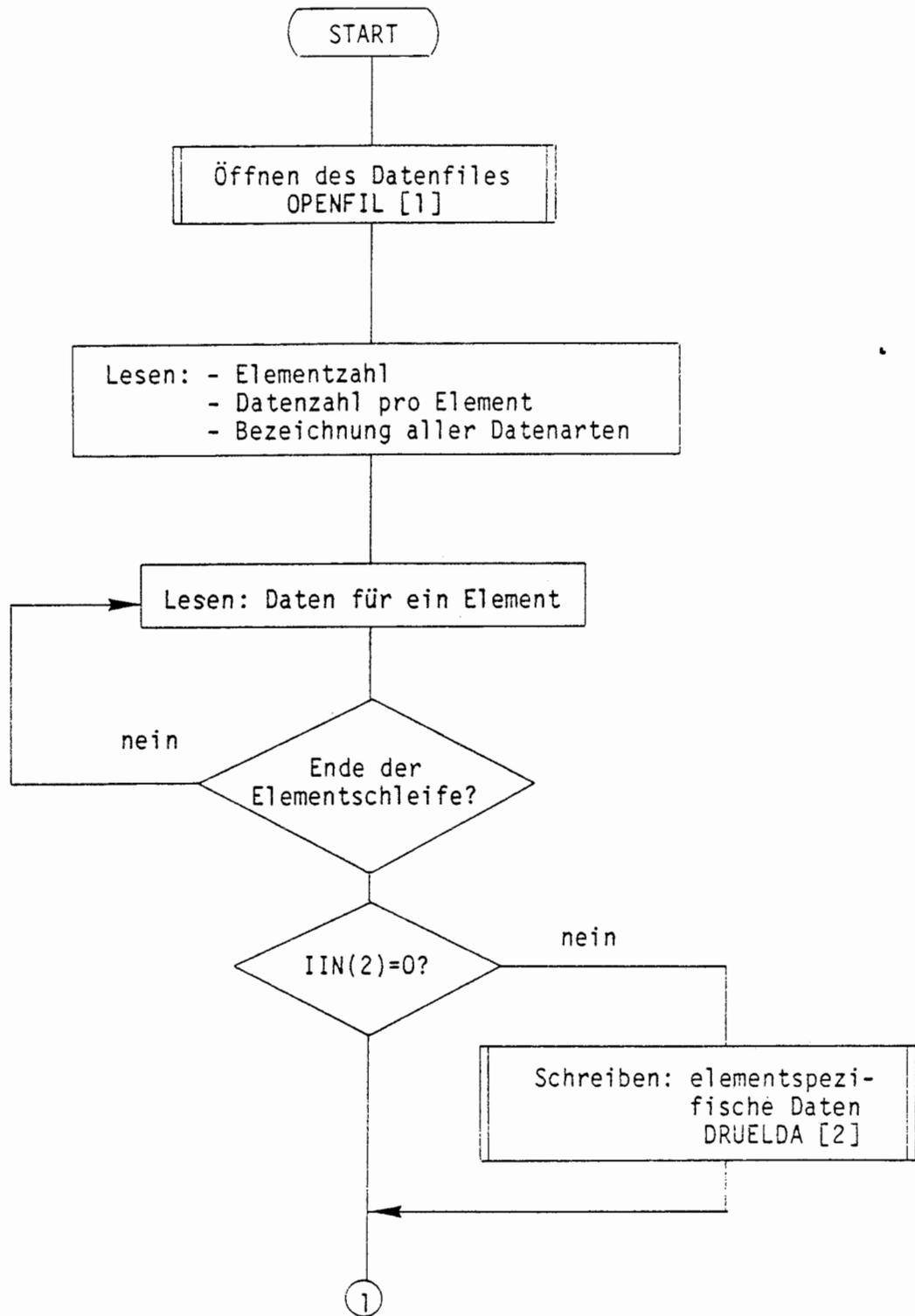
SCHREIBENDER ZUGRIFF AUF COMMON-BLÖCKE:

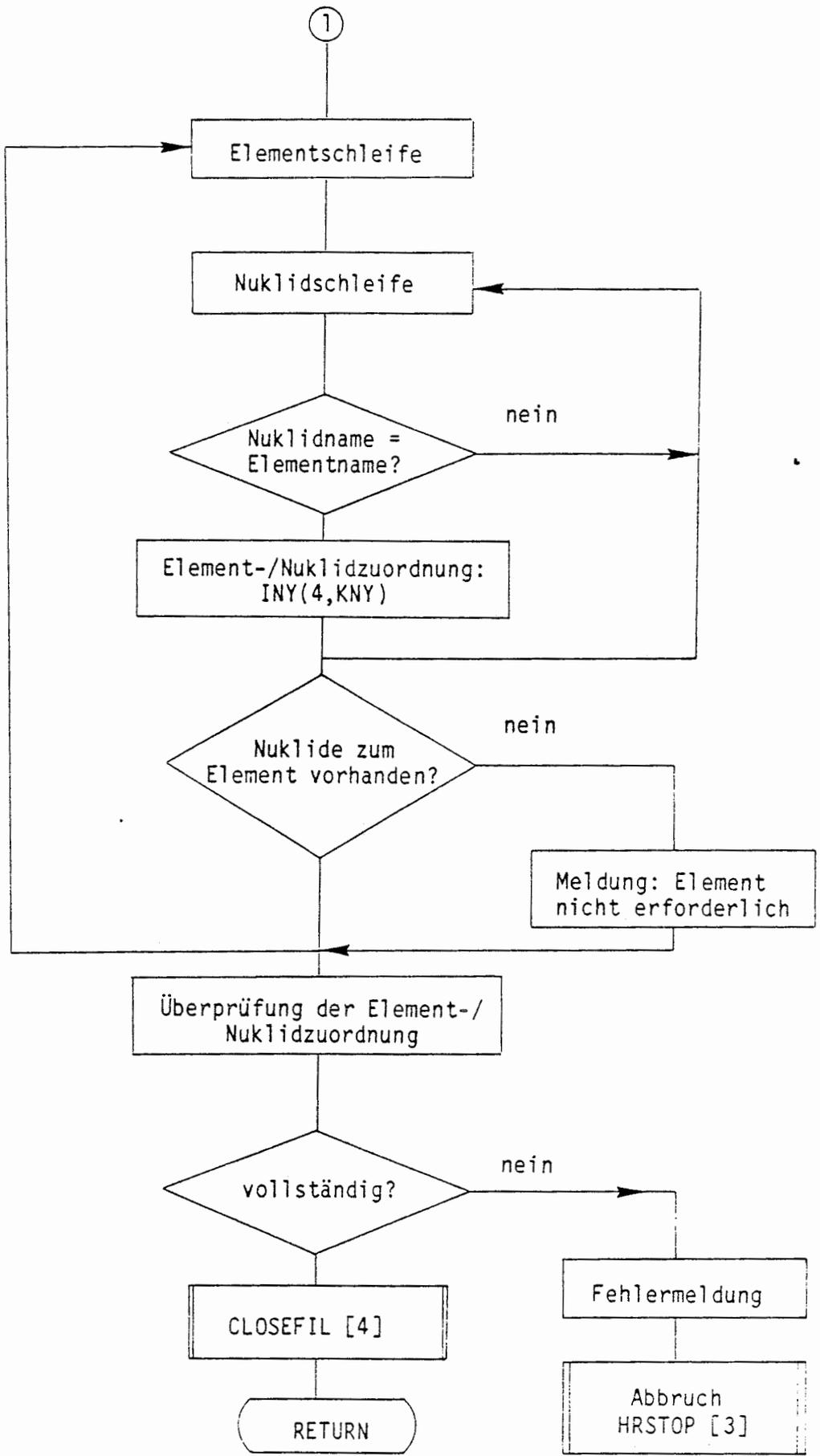
/WZEY/
/WCEY/ : CEY
/WZNY/ : INY(4, KNY)

UNTERPROGRAMMAUFRUFE:

- [1] OPENFIL (KANAL, 'ELDA')
- [2] DRUELDA
- [3] HRSTOP ('ELDAIN')
- [4] CLOSEFIL ('ELDA')

ABLAUF DES PROGRAMMS ELDAIN





A 2.4.2.2 AUSGABE ELEMENTSPEZIFISCHER DATEN (DRUELDA)

DRUELDA schreibt die elementspezifischen Daten in das Output-File. Die Daten für alle Elemente werden jeweils nach Datenart (wie z.B. Löslichkeitsgrenzen, KD-Werte und Mobilisierungsdauern) unterteilt und geschrieben.

ÜBERSICHT ÜBER DAS PROGRAMM DRUELDA

ÜBERGEORDNETES PROGRAMM : DALEKO

LÄNGE DES PROGRAMMS : 81 Zeilen
 178 Bytes

STAND DER DOKUMENTATION : 15. März 1986

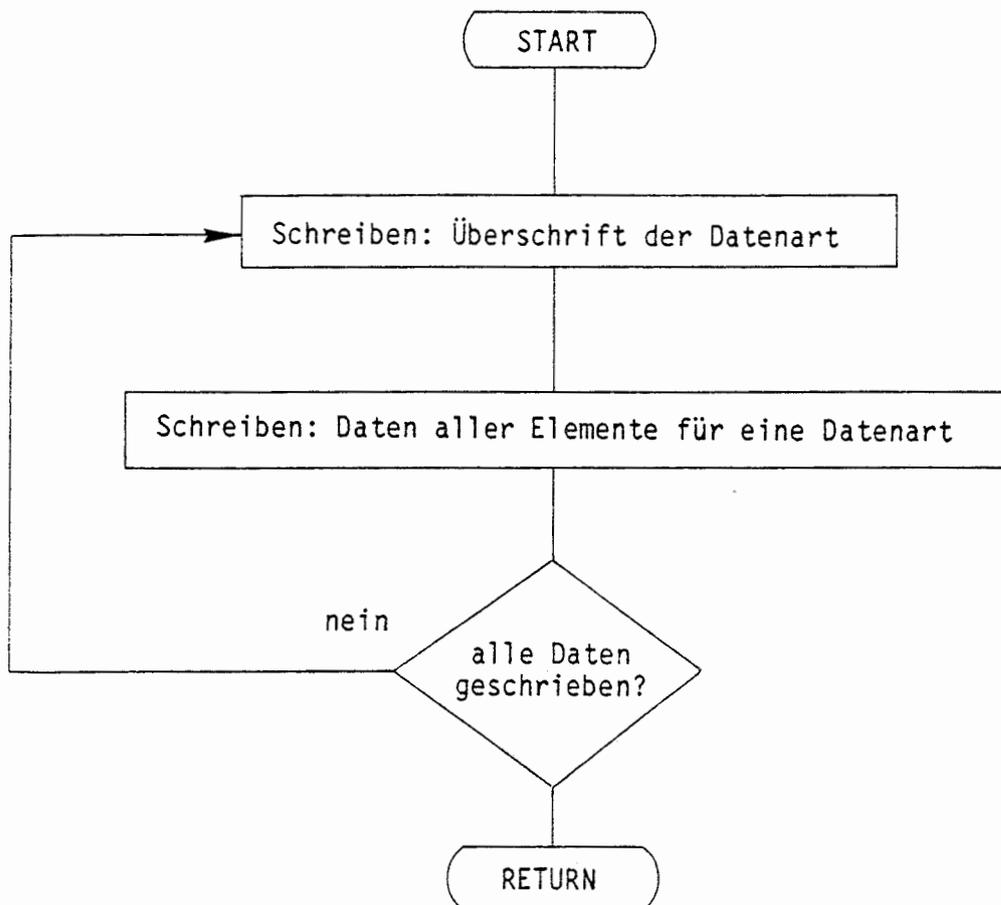
KURZBESCHREIBUNG:

Schreiben elementspezifischer Daten in das Output-File.

LESENDER ZUGRIFF AUF COMMON-BLÖCKE:

/WCEY/, /WZEY/

ABLAUF DES PROGRAMMS DRUELDA



A 2.4.3 EIN-/AUSGABE VON STRUKTURDATEN (SDATIN)

Die zu untersuchende Barrierenstruktur wird durch Angaben im Job-Input-File festgelegt. Hier wird für jede Barriere angegeben, welche anderen Barrieren Radionuklide an die betrachtete Barriere abgeben können. Die Angabe solcher Verknüpfungspunkte in der Barrierenstruktur ist für alle Teilbereiche des Grubengebäudes, nur nicht für Abfallgebinde, erforderlich.

Das Programm SDATIN liest die Barrierennamen für alle Verknüpfungspunkte in die Strukturmatrix (CSX(LISY-1, LSY). Ein jeder Verknüpfungspunkt wird durch eine Zeile im Job-Input-File definiert und in einer Spalte der Strukturmatrix (Strukturvektor) abgelegt. In der ersten Zeile dieser Matrix stehen dann die Namen der aufnehmenden Barriere. Die weiteren Elemente der Spalten enthalten dann die Namen der Barrieren, die Radionuklide an die aufnehmende Barriere abgeben können.

Nach der Eingabe der Strukturmatrix werden für jede Spalte die Anzahl der Eingangsbarrieren ermittelt und in der ersten Zeile der Matrix ISY(1, NSY) abgelegt.

Nach der Übernahme eines Strukturvektors wird dieser in das Output-File geschrieben. Falls die Zahl der Strukturvektoren größer als die entsprechende Dimensionierung der Strukturmatrix wird, kommt es zum Programmabbruch.

ÜBERSICHT ÜBER DAS PROGRAMM SDATIN

ÜBERGEORDNETES PROGRAMM :	DALEKO
LÄNGE DES PROGRAMMS :	149 Zeilen 519 Bytes
STAND DER DOKUMENTATION :	15. März 1986

KURZBESCHREIBUNG:

Übernahme und Ausdruck der Barrierenstruktur (Strukturdaten)

LESENDER ZUGRIFF AUF COMMON-BLÖCKE:

/WZSY/ : NSY
/WTEXT/ : CBLANK

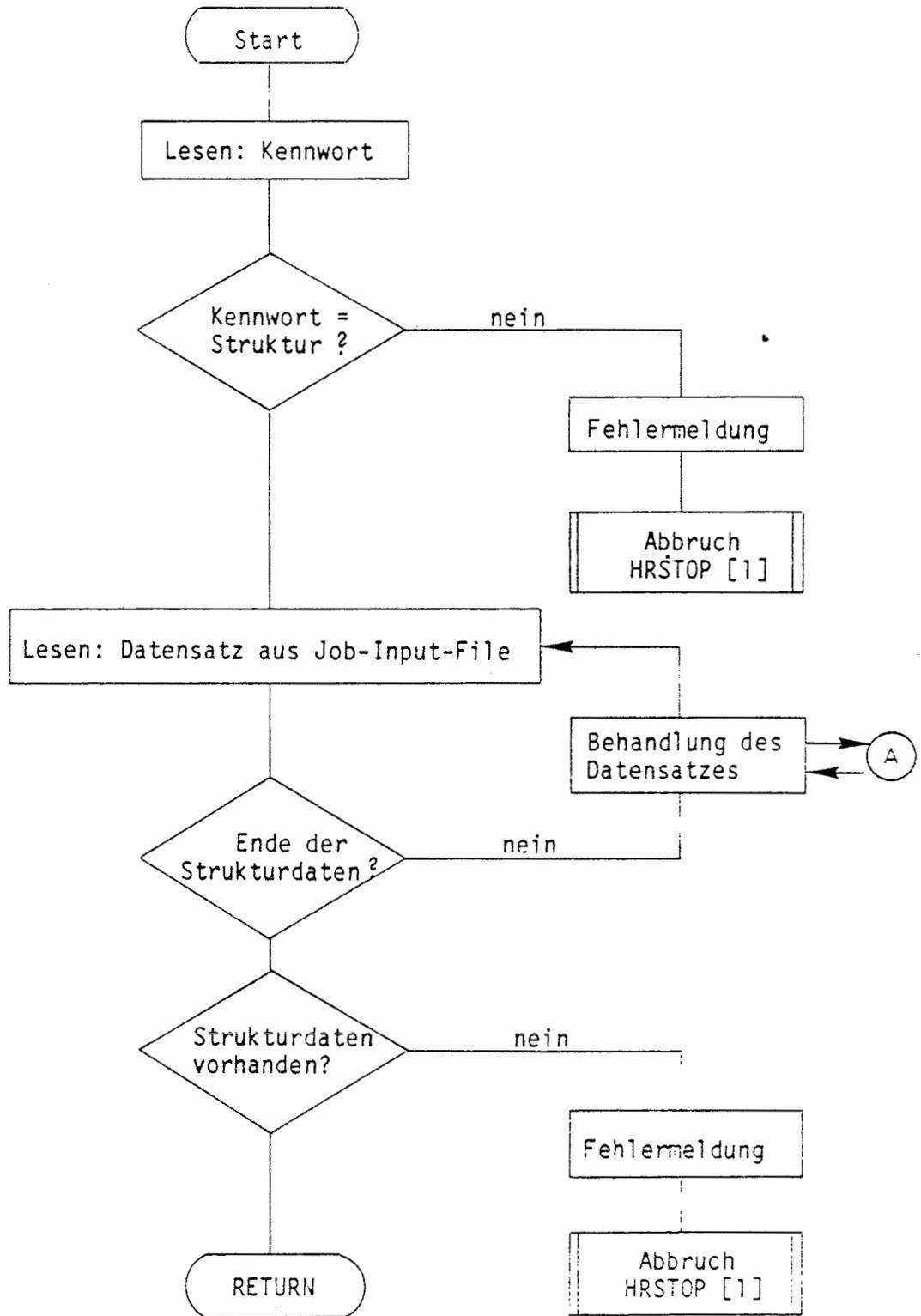
SCHREIBENDER ZUGRIFF AUF COMMON-BLÖCKE:

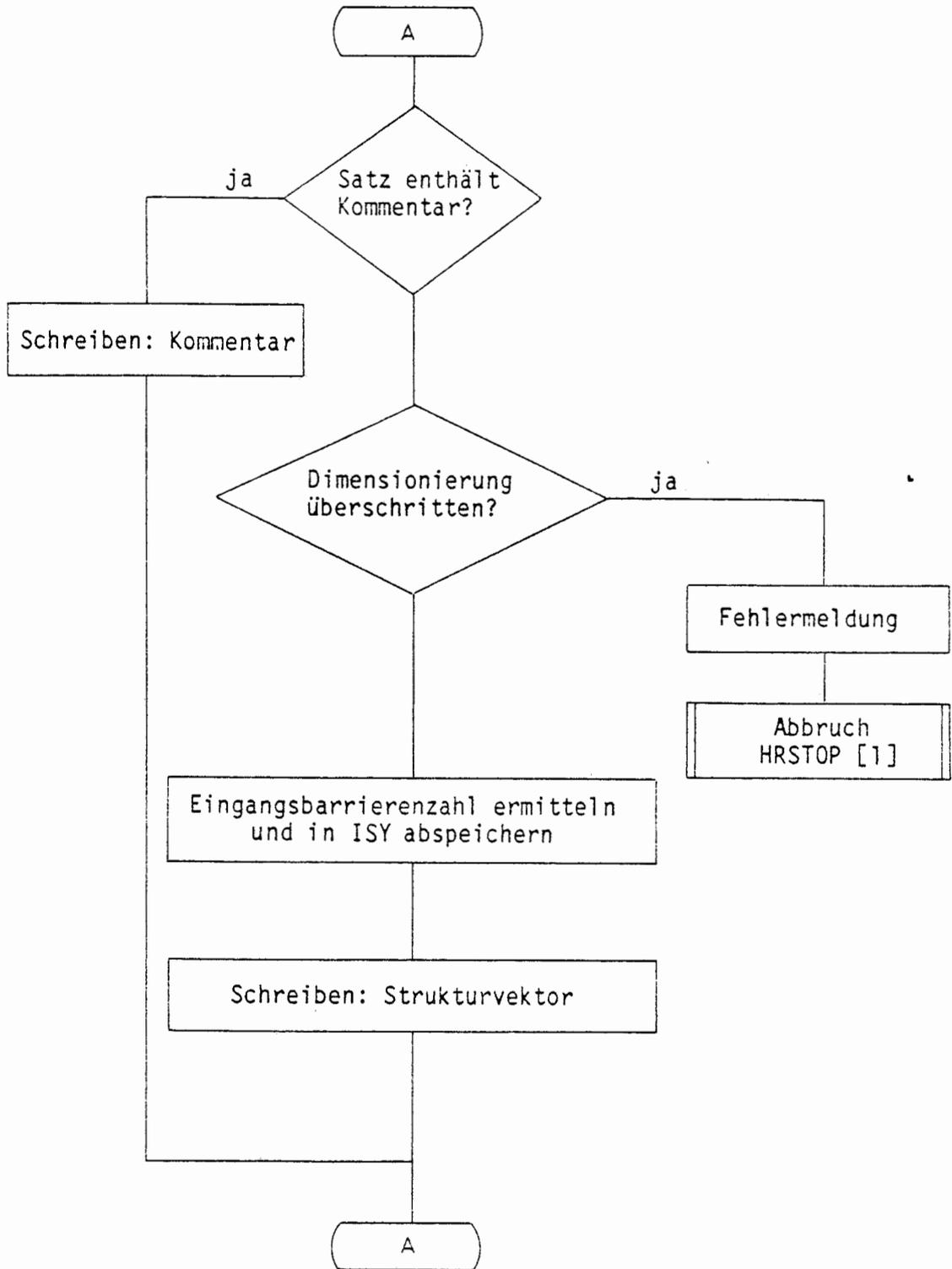
/WZSY/ : NSY, ISY(1, NSY)
/WCSX/

UNTERPROGRAMMAUFRUFE:

[1] HRSTOP ('SDATIN')

ABLAUF DES PROGRAMMS SDATIN





A 2.4.4 BARRIERENSPEZIFISCHE DATEN (BDATIN)

Zur Bereitstellung der Eingangsdaten für die Barrierenmodelle aller Barrieren werden sogenannte globale Barrierendaten und barrierenspezifische Daten übernommen. Die globalen Barrierendaten beinhalten physikalische und chemische Konstanten, die in allen Barrierenmodellen einheitlich verwendet werden. Die barrierenspezifischen Daten betreffen physikalische, geometrische oder andere Konstanten, die für eine einzelne Barriere zutreffend sind. Der Umfang und der Inhalt eines barrierenspezifischen Datensatzes richtet sich nach der Art des zugehörigen Barrierenmodells.

Zur Eingabe der Barrierendaten werden als erstes die globalen Barrierendaten vom Job-Input-File gelesen und durch das Programm PRTGLB ausgedruckt, wobei der Inhalt der einzelnen Positionen erläutert wird. Danach werden bei entsprechender Ansteuerung durch den Parameter IST(4) die barrierenspezifischen Datensätze eines Files der Bibliothek durch das Programm BADAIN eingelesen.

Die von der Bibliothek übernommenen Datensätze können dann über das Job-Input-File modifiziert und ergänzt werden. Hierbei wird jeweils ein barrierenspezifischer Datensatz in einen Zwischenspeicherbereich eingelesen. Nach Überprüfung der Existenz der eingelesenen Barriere in der Barrierenstruktur durch das Programm PRBAST wird der Datensatz in den barrierenspezifischen Speicherbereich übertragen (Programm BMIN). Je nachdem, ob eine entsprechende Barriere bereits vorhanden ist, wird entweder der barrierenspezifische Speicherbereich erweitert und der neue Datensatz angehängt oder der Datensatz für die vorhandene Barriere überschrieben.

Der noch im Zwischenspeicherbereich vorhandene Datensatz wird bei Bedarf und Ansteuerung über den Parameter IIN(5) durch das Programm DRUBADA angegeben.

Nach Abschluß der Eingabe barrierenspezifischer Datensätze sind die Barrierenmatrizen, die in den COMMON-Blöcken /WCBY/ und /WZBY/ abgelegt sind, in wesentlichen Teilen vorhanden. Falls barrierenspezifische INTEGER- oder REAL-Größen für eine Barriere nicht angegeben wurden, sondern von einer anderen Barriere zu übernehmen sind, müssen noch die entsprechenden Kopiervorgänge durchgeführt werden. Weitere Ergänzungen finden nach der Übernahme der abfallspezifischen Daten und innerhalb der zeitdiskreten Rechnung statt.

Während der Übernahme eines Datensatzes und insbesondere nach der Übertragung aller Datensätze in die Barrierenmatrizen werden verschiedene Fehlerkontrollen durchgeführt:

- Überschreitet die Anzahl der Datensätze die Dimensionierung des zugehörigen Speicherbereichs?
- Ist die Barriere in der Barrierenstruktur vorhanden?
- Ist eine gewünschte Übernahme bestimmter Teilmengen der barrierenspezifischen Daten von anderen Barrieren möglich?
- Sind barrierenspezifische Daten für alle Barrieren der Barrierenstruktur vorhanden?

Falls bei einer der Kontrollen ein Fehler entdeckt wird, findet ein Programmabbruch entweder unmittelbar oder nach Fertigstellung aller Kontrollen statt.

ÜBERSICHT ÜBER DAS PROGRAMM BDATIN

ÜBERGEORDNETES PROGRAMM :	DALEKO
LÄNGE DES PROGRAMMS :	331 Zeilen 1347 Bytes
STAND DER DOKUMENTATION :	15. März 1986

KURZBESCHREIBUNG:

Eingabe globaler Barrierendaten sowie Eingabe barrierenspezifischer Daten und Aufbau der Barrierenmatrizen.

LESENDER ZUGRIFF AUF COMMON-BLÖCKE:

/ / : IST(4)
/RZ/ : IIN(5)
/WCB1X/, /WZBX/, /WCBX/
/WZNY/, /WZBY/

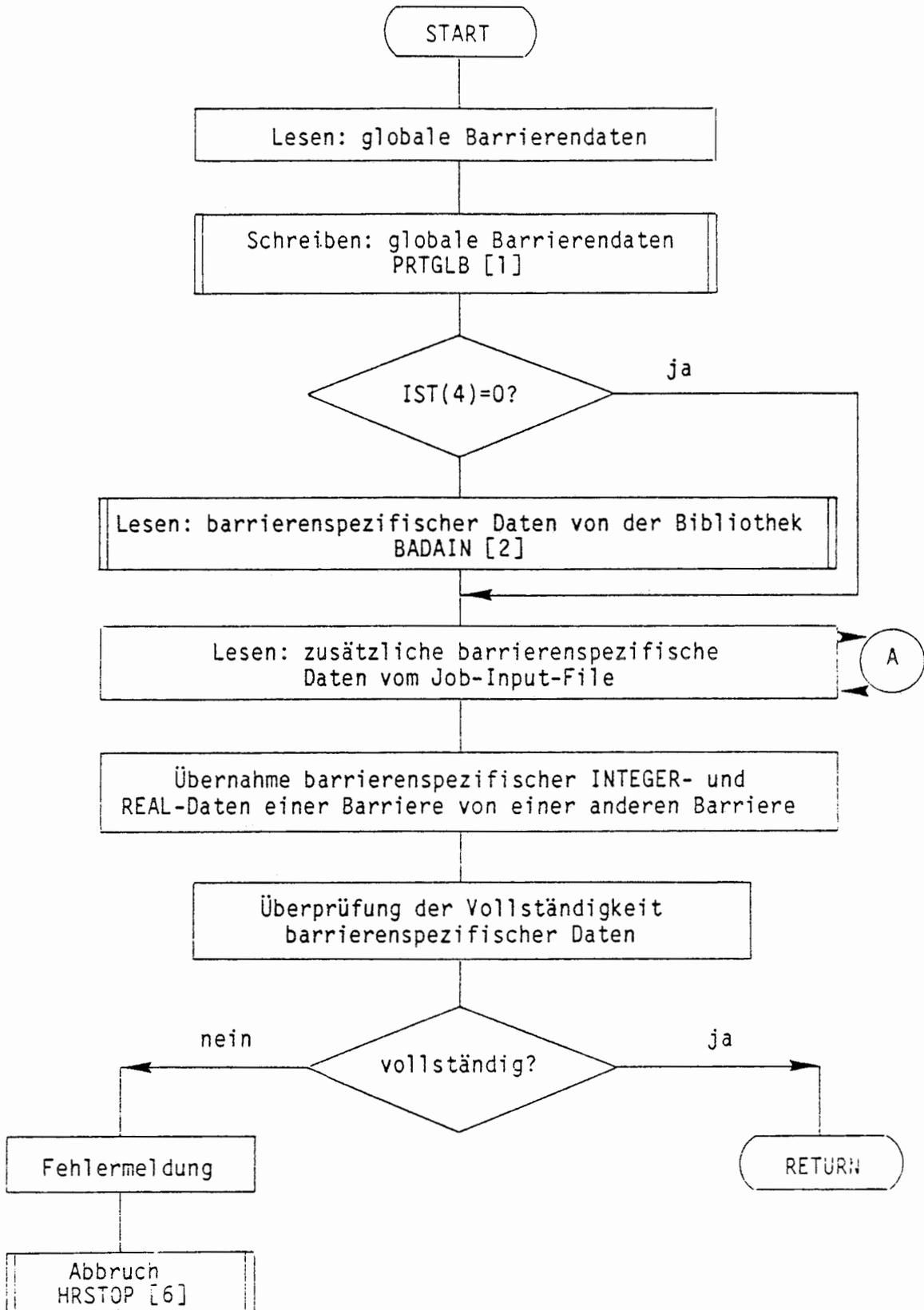
SCHREIBENDER ZUGRIFF AUF COMMON-BLÖCKE:

/G/ : GID
/WCB1X/, /WZBX/, /WZBY/

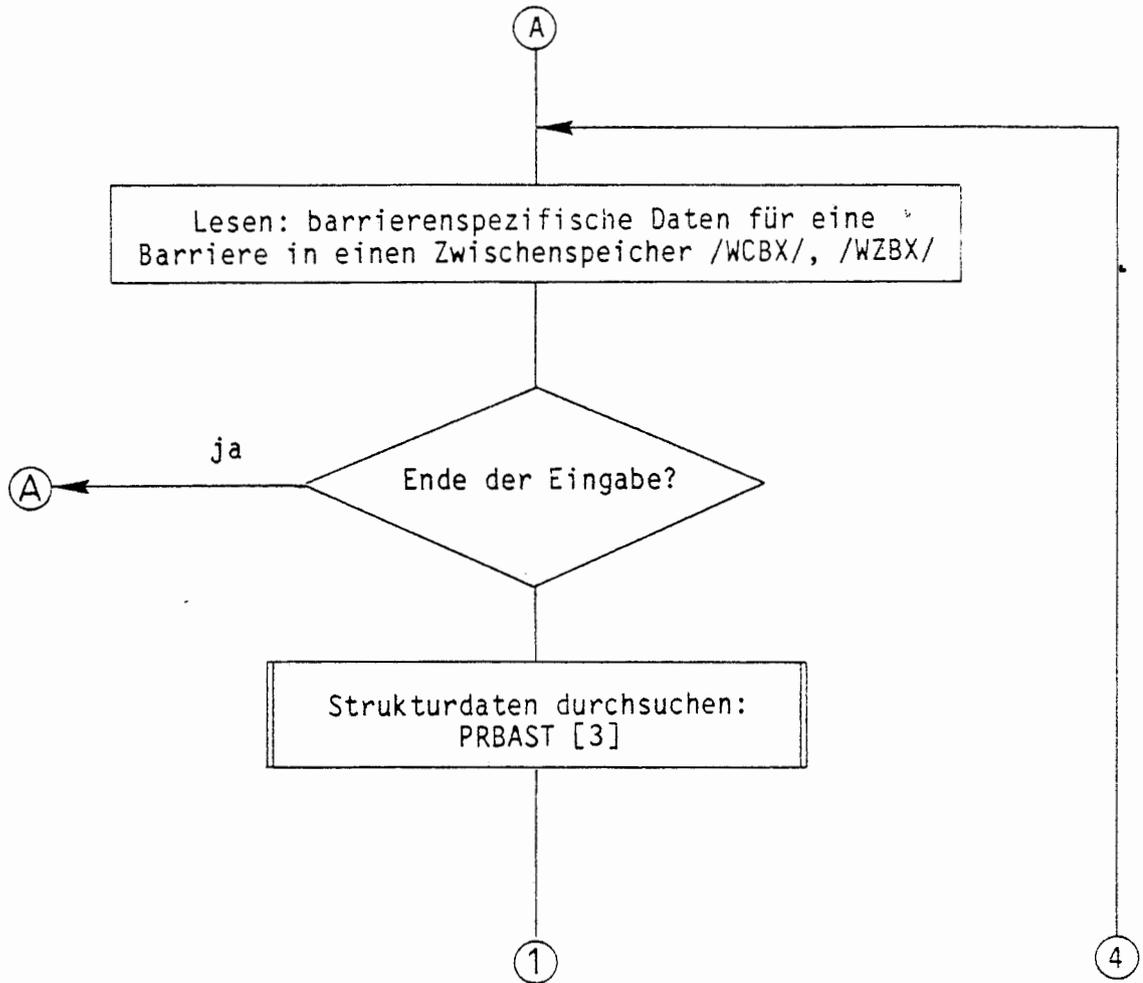
UNTERPROGRAMMAUFRUFE:

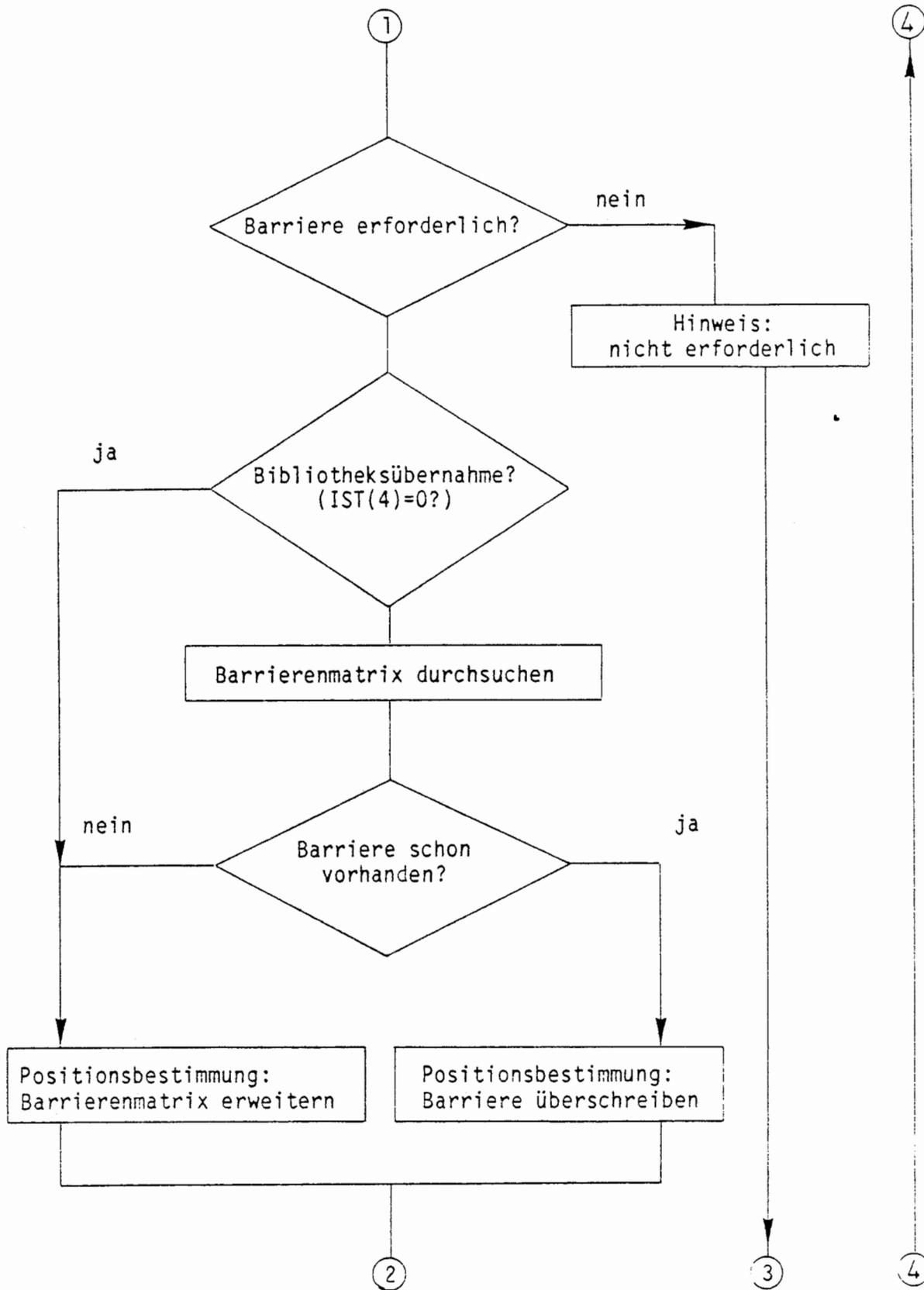
[1] PRTGLB
[2] BADATIN(IST(4))
[3] PRBAST
[4] BMIN
[5] DRUBADA
[6] HRSTOP('BDATIN')

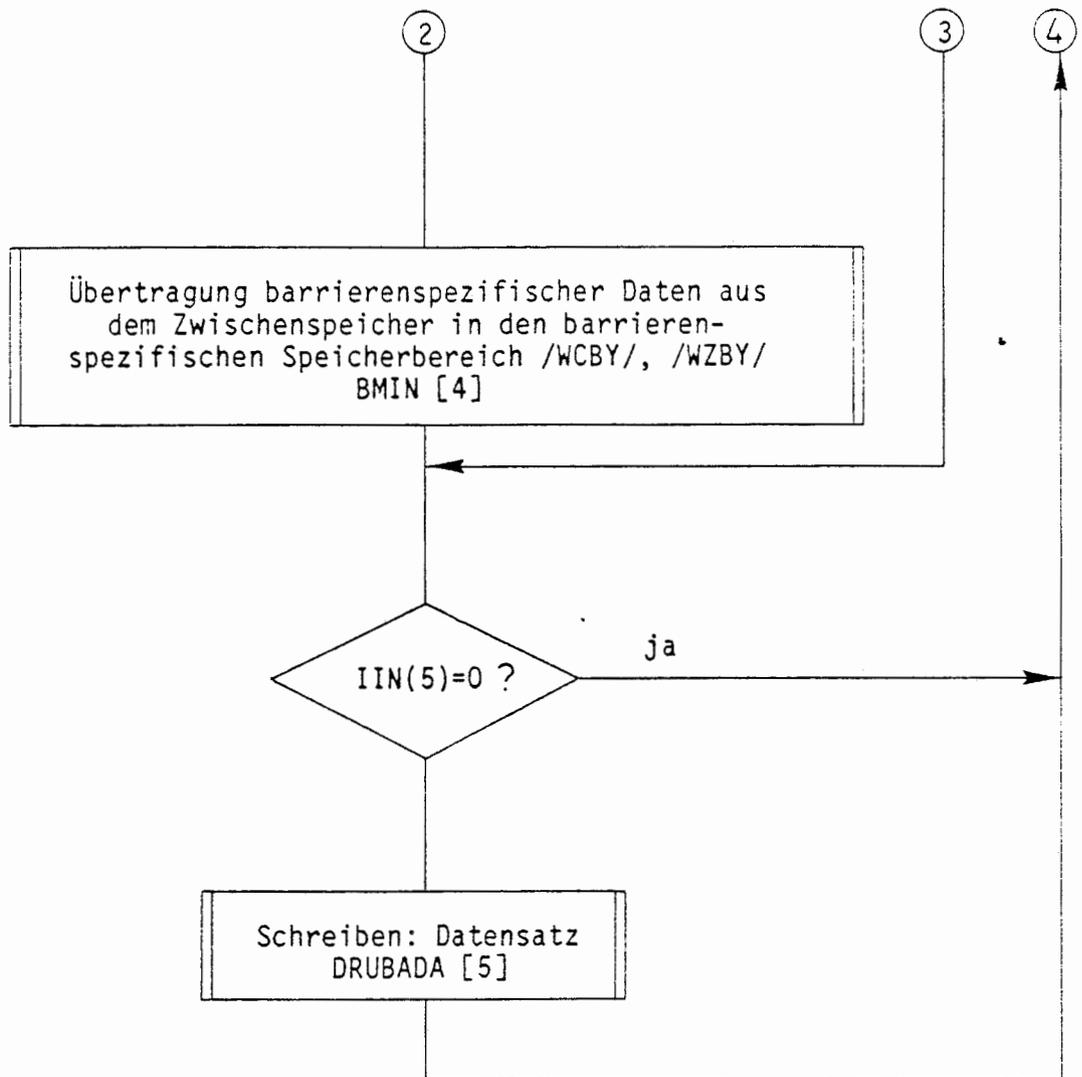
ABLAUF DES PROGRAMMS BDATIN



ABLAUF DES PROGRAMMS BDATIN TEILAUFGABE A







A 2.4.4.1 AUSGABE GLOBALER BARRIERENDATEN (PRTGLB)

Das Programm PRTGLB dient zum Ausdrucken der globalen Barrierendaten. Für jede Position des belegten Speicherbereiches wird der Zahlenwert und die Bedeutung der Position ausgegeben.

ÜBERSICHT ÜBER DAS PROGRAMM PRTGLB

ÜBERGEORDNETES PROGRAMM : DALEKO

LÄNGE DES PROGRAMMS : 89 Zeilen
105 Bytes

STAND DER DOKUMENTATION : 15. März 1986

KURZBESCHREIBUNG:

Ausdruck globaler Barrierendaten

EINGANGSPARAMETER:

K1 : Anzahl der Positionen des belegten Speicherbereich

LESENDER ZUGRIFF AUF COMMON-BLÖCKE:

/G/

Da der Programmtext im wesentlichen aus Schreibanweisungen besteht, wird auf die Darstellung eines Ablaufdiagramms verzichtet.

A 2.4.4.2 EINGABE BARRIERENSPEZIFISCHER DATEN (BADAIN)

Das Programm BADAIN liest alle barrierenspezifischen Datensätze aus einem Datenfile der Bibliothek. Nach der Übernahme eines Datensatzes in den Zwischenspeicher wird durch das Programm PRBAST die Existenz dieser Barriere in der Barrierenstruktur überprüft. Danach erfolgt ggf. eine Übertragung in den barrierenspezifischen Speicherbereich durch das Programm BMIN, wobei dieser Bereich durch die Aufnahme einer neuen Barriere entsprechend erweitert wird. Der noch im Zwischenspeicherbereich vorhandene Datensatz wird bei Bedarf und Ansteuerung über den Parameter IIN(5) durch das Programm DRUBADA ausgegeben.

Bei der Übernahme eines barrierenspezifischen Datensatzes werden zunächst die allgemeinen barrierenspezifischen Daten folgender Art eingelesen:

CB1X(1)	:	Name der Barriere
CB1X(2)	:	Name des Barrierenmodells
CB1X(3)	:	Name des eingelagerten Abfalls
CB1X(4)	:	Name einer Barriere zur Datenübernahme
IB1X(1)	:	Wiederholungsfaktor der Barriere
IB1X(2)	:	Gebindezahl in der Abfallmischung
RB1X(1)	:	Einlagerungszeitpunkt
RB1X(2)	:) ohne Bedeutung bei der Eingabe, Belegung bis)
RB1X(4)	:) in der zeitdiskreten Rechnung

Falls keine Datenübernahme von einer anderen Barriere gewünscht wird (CB1X(4)='') werden die barrierenspezifischen INTEGER-Daten und die barrierenspezifischen REAL-Daten auf die Zwischenspeicherbereiche IB2X und RB2X eingelesen.

Vor der Übernahme eines Datensatzes in den barrierenspezifischen Speicherbereich wird überprüft, ob die Anzahl der Datensätze die Dimensionierung des Speicherbereichs überschreitet.

ÜBERSICHT ÜBER DAS PROGRAMM BADAIN

ÜBERGEORDNETES PROGRAMM : DALEKO
LÄNGE DES PROGRAMMS : 171 Zeilen
400 Bytes
STAND DER DOKUMENTATION : 15. März 1986

KURZBESCHREIBUNG:

Lesen barrierenspezifischer Daten aus der Datenbibliothek.

PARAMETERLISTE:

KANAL

EINGANGSPARAMETER:

KANAL : Nummer des Datenfiles

LESENDER ZUGRIFF AUF COMMON-BLÖCKE:

/WZBY/ : NBY
/WTEXT/ : CBLANK
/RZ/ : IIN(5)

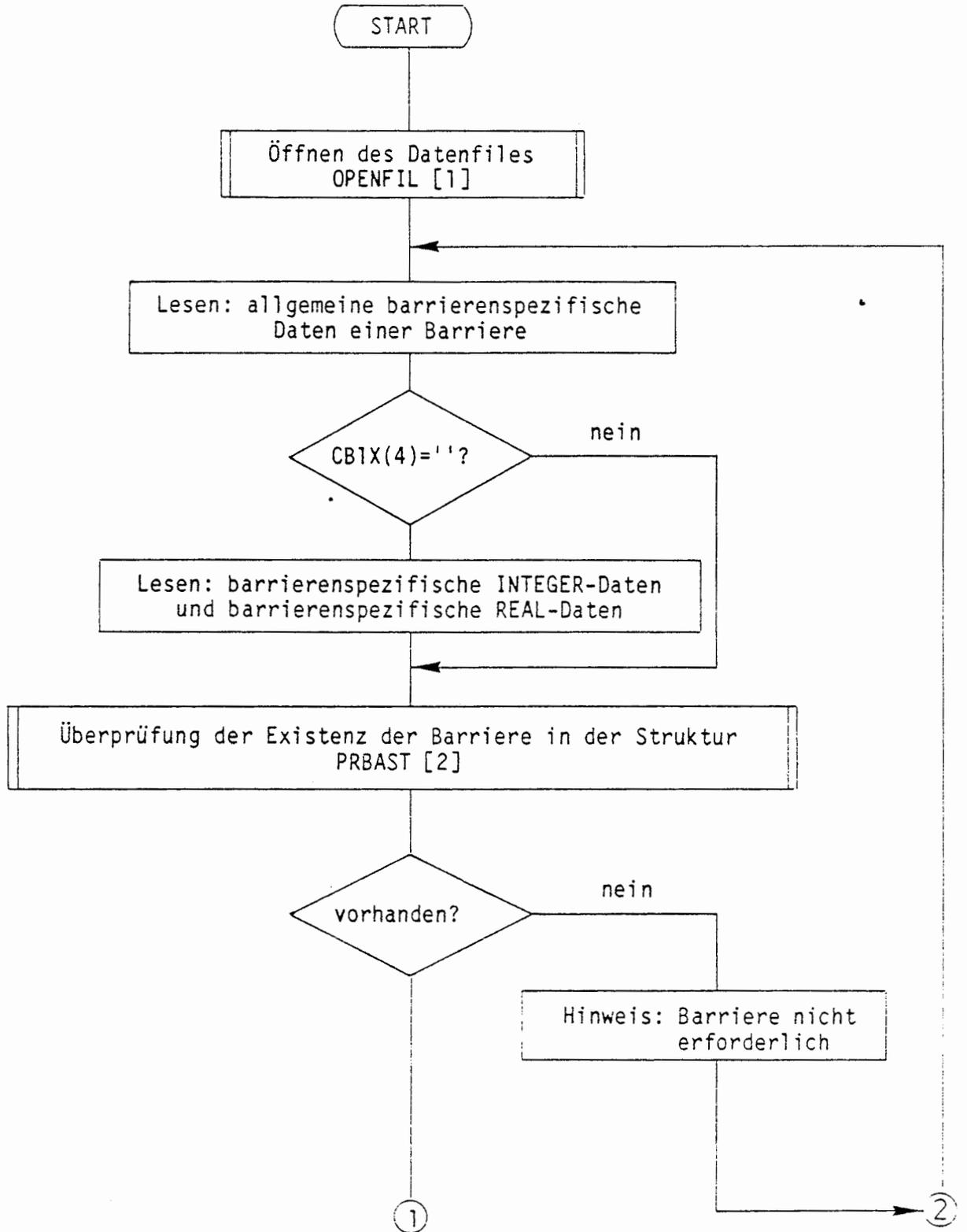
SCHREIBENDER ZUGRIFF AUF COMMON-BLÖCKE:

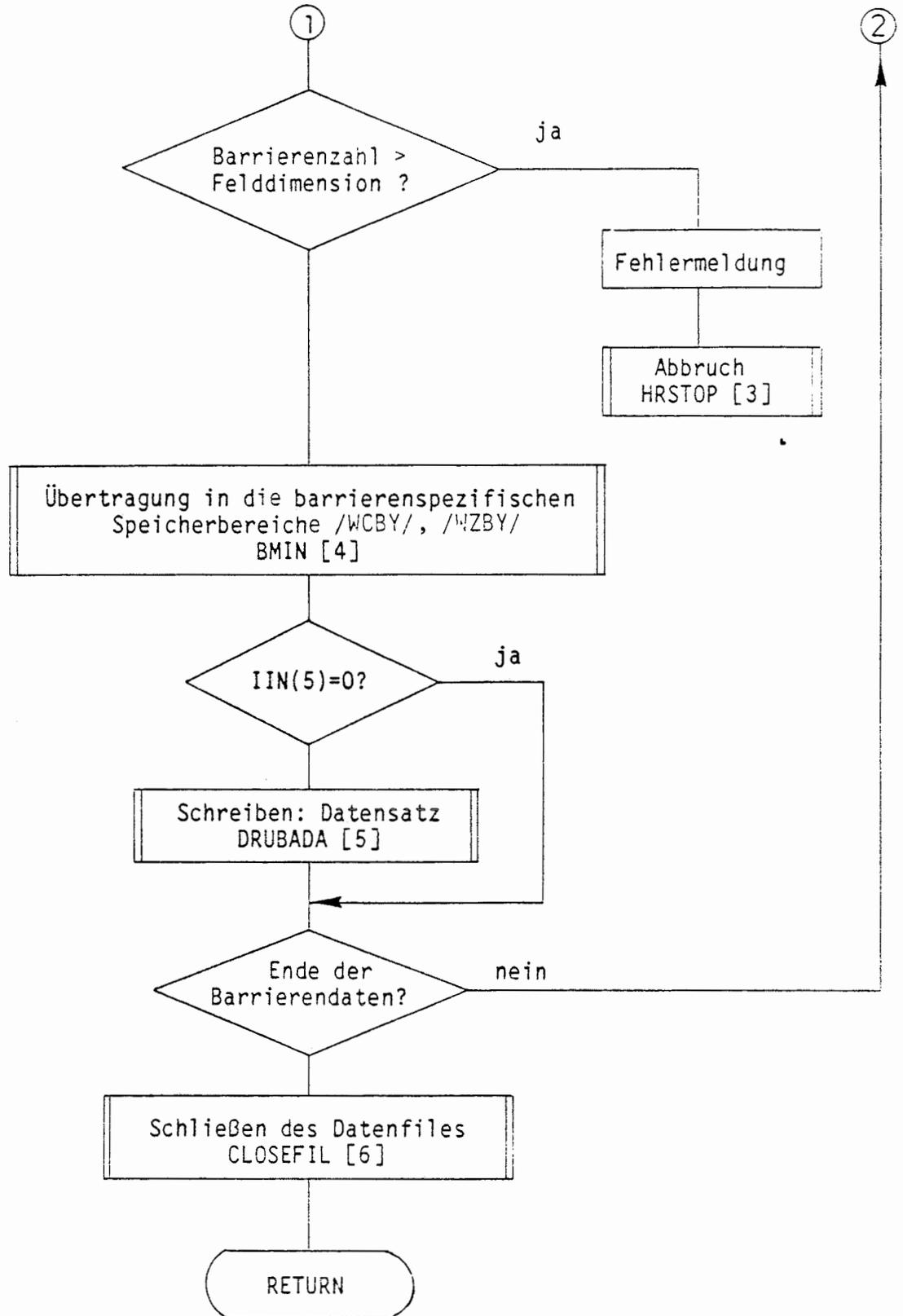
/WZBX/, /WCB1X/

UNTERPROGRAMMAUFRUFE:

- [1] OPENFIL (KANAL, 'BADA')
- [2] PRBAST (BAIST)
- [3] HRSTOP ('BADAIN')
- [4] BMIN (NBY)
- [5] DRUBADA (IUEBS)
- [6] CLOSEFIL ('BADA')

ABLAUF DES PROGRAMMS BADAIN





A 2.4.4.3 ÜBERPRÜFUNG EINER BARRIERE (PRBAST)

Das Programm PRBAST prüft, ob der Name der Barriere, deren Datensatz gerade im Zwischenspeicherbereich für barrierenspezifische Daten abgelegt ist, in der Strukturmatrix vorhanden ist. Die Strukturmatrix im COMMON-Block /WCSX/ wird spaltenweise nach dem Barrierennamen abgesehen. Das Programm endet mit BAIST='TRUE', wenn der Barrierename gefunden wurde, ansonsten mit BAIST='FALSE'.

ÜBERSICHT ÜBER DAS PROGRAMM PRBAST

ÜBERGEORDNETES PROGRAMM : DALEKO

LÄNGE DES PROGRAMMS : 108 Zeilen
 110 Bytes

STAND DER DOKUMENTATION : 15. März 1986

KURZBESCHREIBUNG:

Prüfung, ob der Name der Barriere, deren Datensatz gerade im Zwischenspeicherbereich der barrierenspezifischen Daten abgelegt ist, in der Strukturmatrix vorhanden ist.

PARAMETERLISTE:

BAIST

AUSGANGSPARAMETER:

BAIST Kennung, ob der Barrierename in der Strukturmatrix vorhanden ist oder nicht.

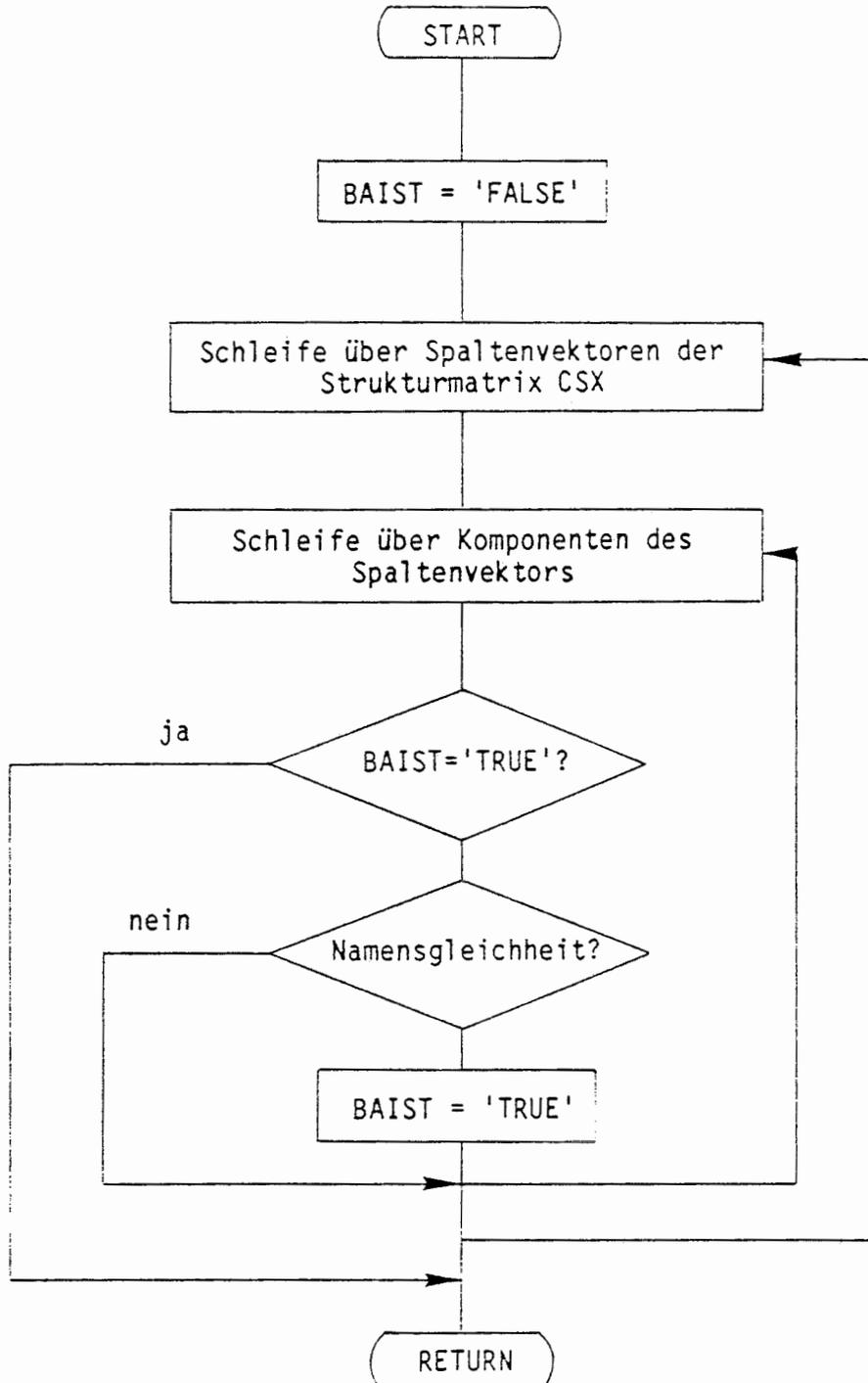
LESENDER ZUGRIFF AUF COMMON-BLÖCKE:

/WZSY/, /WCSX/, /WCB1X/: CB1X(1)

SCHREIBENDER ZUGRIFF AUF COMMON-BLÖCKE:

/WZSY/ : NISY

ABLAUF DES PROGRAMMS PRBAST



A 2.4.4.4 ÜBERNAHME FÜR EINE BARRIERE (BMIN)

Durch das programm BMIN wird ein barrierenspezifischer Datensatz aus einem Zwischenspeicherbereich in den Common-Blöcken /WCB1X/ und /WZBX/ in den barrierenspezifischen Speicherbereich (Barrierenmatrix) in den Common-Blöcken /WCBY/ und /WZBY/ übertragen. Der Datensatz wird hierbei in die angegebene Spaltennummer der Barrierenmatrix geschrieben. Falls dort bereits ein Datensatz für eine Barriere abgelegt ist, wird dieser überschrieben.

ÜBERSICHT ÜBER DAS PROGRAMM BMIN

ÜBERGEORDNETES PROGRAMM : DALEKO

LÄNGE DES PROGRAMMS : 107 Zeilen
 240 Bytes

STAND DER DOKUMENTATION : 15. März 1986

KURZBESCHREIBUNG:

Übertragung eines barrierenspezifischen Datensatzes aus dem Zwischenspeicherbereich in die Barrierenmatrix

EINGANGSPARAMETER:

KBX : Nummer der Spalte der Barrierenmatrix

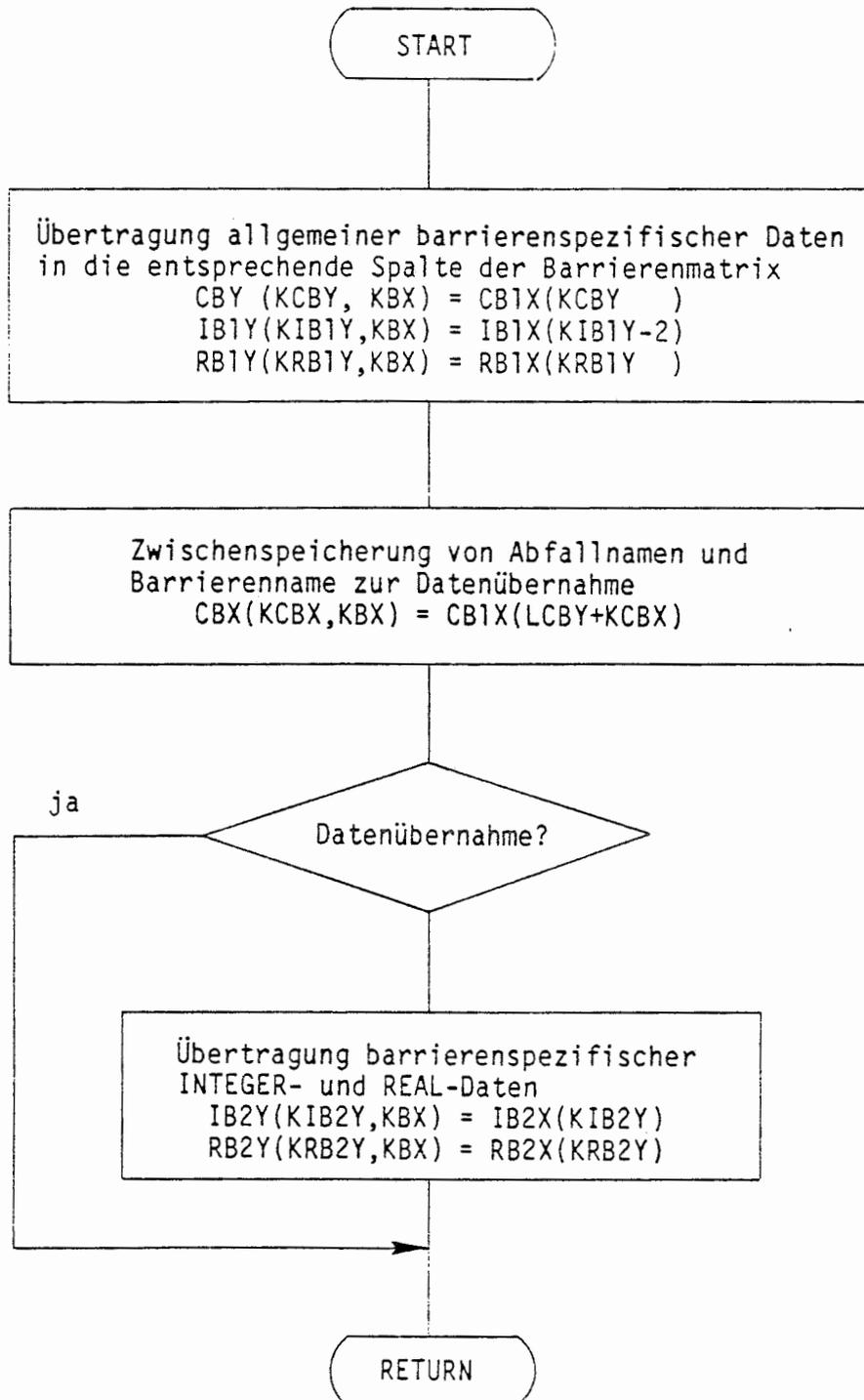
LESENDER ZUGRIFF AUF COMMON-BLÖCKE:

/WCB1X/, /WZBX/, /WTEXT/

SCHREIBENDER ZUGRIFF AUF COMMON-BLÖCKE:

/WCBY/, /WCBX/, /WZBY/

ABLAUF DES PROGRAMMES BMIN



A 2.4.4.5 AUSGABE BARRIERENSPEZIFISCHER DATEN (DRUBADA)

Das Programm DRUBADA schreibt die barrierenspezifischen Daten derjenigen Barriere, die im Zwischenspeicherbereich abgelegt ist. Auf Wunsch kann eine Überschrift ausgegeben werden. Nach der Ausgabe der allgemeinen barrierenspezifischen Daten wird geprüft, ob für die betrachtete Barriere ein Barrierenname zur Datenübernahme vorhanden ist. Falls nicht, werden die barrierenspezifischen INTEGER- und REAL-Daten geschrieben.

ÜBERSICHT ÜBER DAS PROGRAMM DRUBADA

ÜBERGEORDNETES PROGRAMM : DALEKO

LÄNGE DES PROGRAMMS : 110 Zeilen
 234 Bytes

STAND DER DOKUMENTATION : 15. März 1986

KURZBESCHREIBUNG:

Ausdrucken eines barrierenspezifischen Datensatzes der Barriere, die aktuell im Zwischenspeicherbereich abgelegt ist.

PARAMETERLISTE:

IUEBS

EINGANGSPARAMETER:

IUEBS : Kennung, ob die Überschrift geschrieben werden soll
oder schon geschrieben ist.

= 0 : Überschrift schreiben

= 1 : Überschrift schon geschrieben

AUSGANGSPARAMETER:

IUEBS

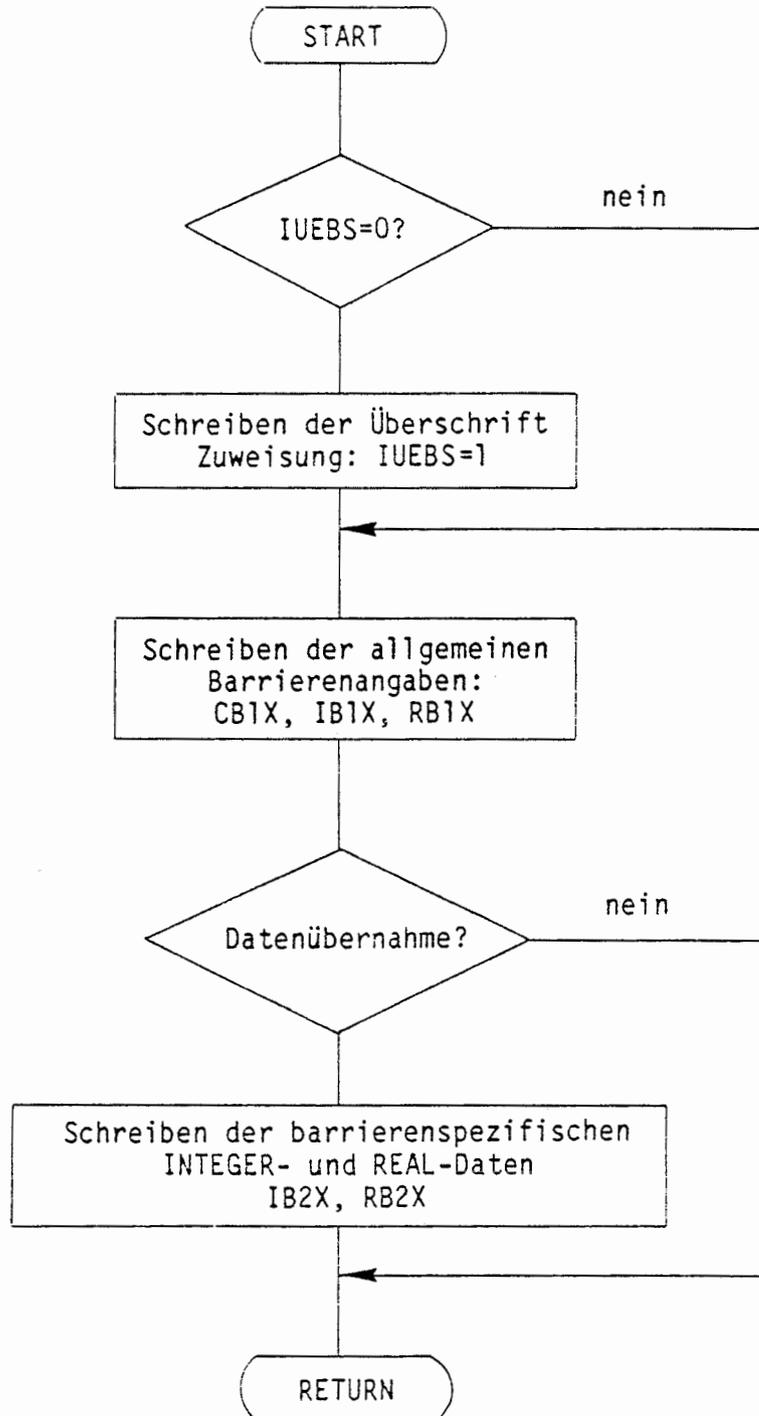
LESENDER ZUGRIFF AUF COMMON-BLÖCKE:

/WCB1X/

/WZBX/

/WTEXT/ : CBLANK

ABLAUF DES PROGRAMMS DRUBADA



A 2.4.5 KODIERUNG DER STRUKTURDATEN (KSEBUE)

Die in der Matrix CSX in Form von Namen angegebene Struktur des Barrierensystems wird im Programm KSEBUE in die Matrix ISY in Form von Integerzahlen umgesetzt. Die Umsetzung in Integerzahlen geschieht so, daß eine eindeutige Zuordnung zwischen Barrierenname und Barriernummer möglich ist. Die Barriernummer entspricht der Reihenfolge bei der Eingabe barrierenspezifischer Datensätze bzw. der Reihenfolge der Ablage in den Barrierenmatrizen. Durch die Kodierung der Barrierennamen wird der Zugriff auf die barrierenspezifischen Datensätze erleichtert.

Bei der Kodierung der Strukturmatrix werden die Strukturvektoren (Spalten der Strukturmatrix) nacheinander bearbeitet. Die erste Komponente dieser Vektoren wurde bereits im Programm SDATIN mit der Zahl der Eingangsbarrieren belegt. Die zweite Komponente erhält die Nummer der aufnehmenden Barriere. Die weiteren Komponenten erhalten die Nummern der Eingangsbarrieren.

Zur Berechnung des Nuklidausgangsstroms einer Barriere bei der zeitdiskreten Rechnung muß der Nuklideingangsstrom bekannt sein. Dies erfordert eine bestimmte Reihenfolge bei der Abarbeitung der Barrieren. Diese wird durch die Reihenfolge der Eingabe barrierenspezifischer Datensätze festgelegt. Inwieweit diese Reihenfolge der oben geschilderten Problematik gerecht wird, kann wie folgt überprüft werden:

Beim Aufbau der Matrix ISY wird zugleich die erste und die zweite Zeile der Integer-Barrierenmatrix IBLY belegt. Enthält die betrachtete Barriere Eingänge, dann wird in die erste Position die Nummer der Barriere und in die zweite Position die Nummer derjenigen Barriere eingetragen, an die die betrachtete Barriere liefert. Durch Vergleich der beiden Zahlenwerte läßt sich eine fehlerhafte Reihenfolge bei der Eingabe barrierenspezifischer Datensätze erkennen. Das Programm bricht ggf. mit einer entsprechenden Fehlermeldung ab.

Während der Kodierung der Strukturmatrix werden Fehlerkontrollen folgenden Inhalts durchgeführt:

- Sind für alle Barrieren des Systems barrierenspezifische Daten eingegeben (siehe Fehlermeldung /1/ im Ablaufdiagramm)?
- Wird eine Barrierenverknüpfung mehrfach definiert (siehe Fehlermeldung /2/ im Ablaufdiagramm)?
- Sind die Barrieren hinsichtlich der Reihenfolge bei der Abarbeitung richtig eingegeben (siehe Fehlermeldung /3/ im Ablaufdiagramm)?
- Ist eine Barriere Eingangsbarriere für verschiedene andere Barrieren (siehe Fehlermeldung /4/ im Ablaufdiagramm)?

Falls bei einer der Kontrollen ein Fehler entdeckt wird, erfolgt ein Programmabbruch.

ÜBERSICHT ÜBER DAS PROGRAMM KSEBUE

ÜBERGEORDNETES PROGRAMM : DALEKO

LÄNGE DES PROGRAMMS : 175 Zeilen
 518 Bytes

STAND DER DOKUMENTATION : 15. März 1986

KURZBESCHREIBUNG:

Kodierung der Strukturmatrix. Ergänzen der Barrierenmatrix. Prüfen der Struktur auf Vollständigkeit der Barrieren.

LESENDER ZUGRIFF AUF COMMON-BLÖCKE:

/WZSY/
/WZBY/ : NBY, IB1Y
/WCBY/ : CBY
/WCSX/ : CSX

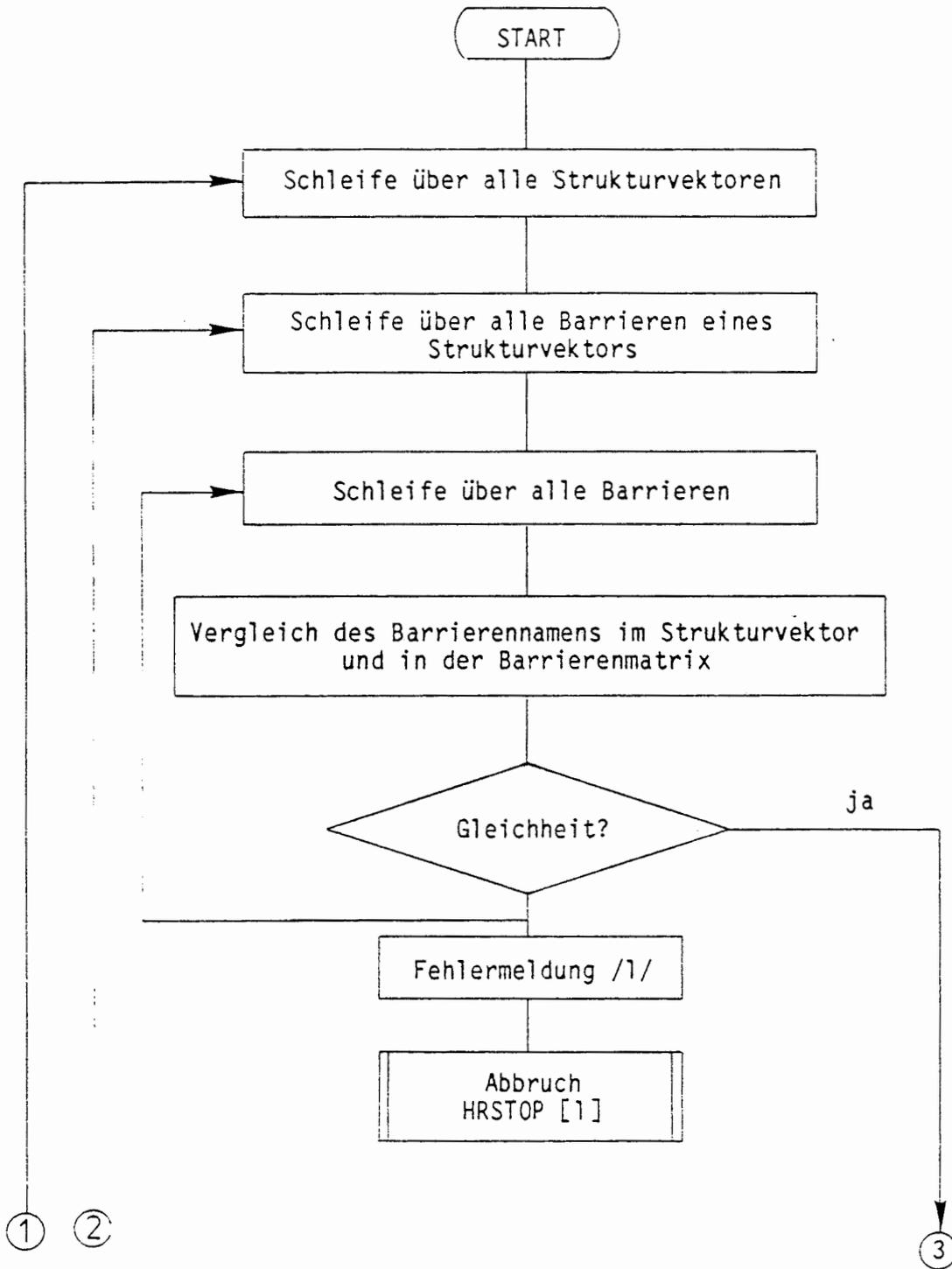
SCHREIBENDER ZUGRIFF AUF COMMON-BLÖCKE:

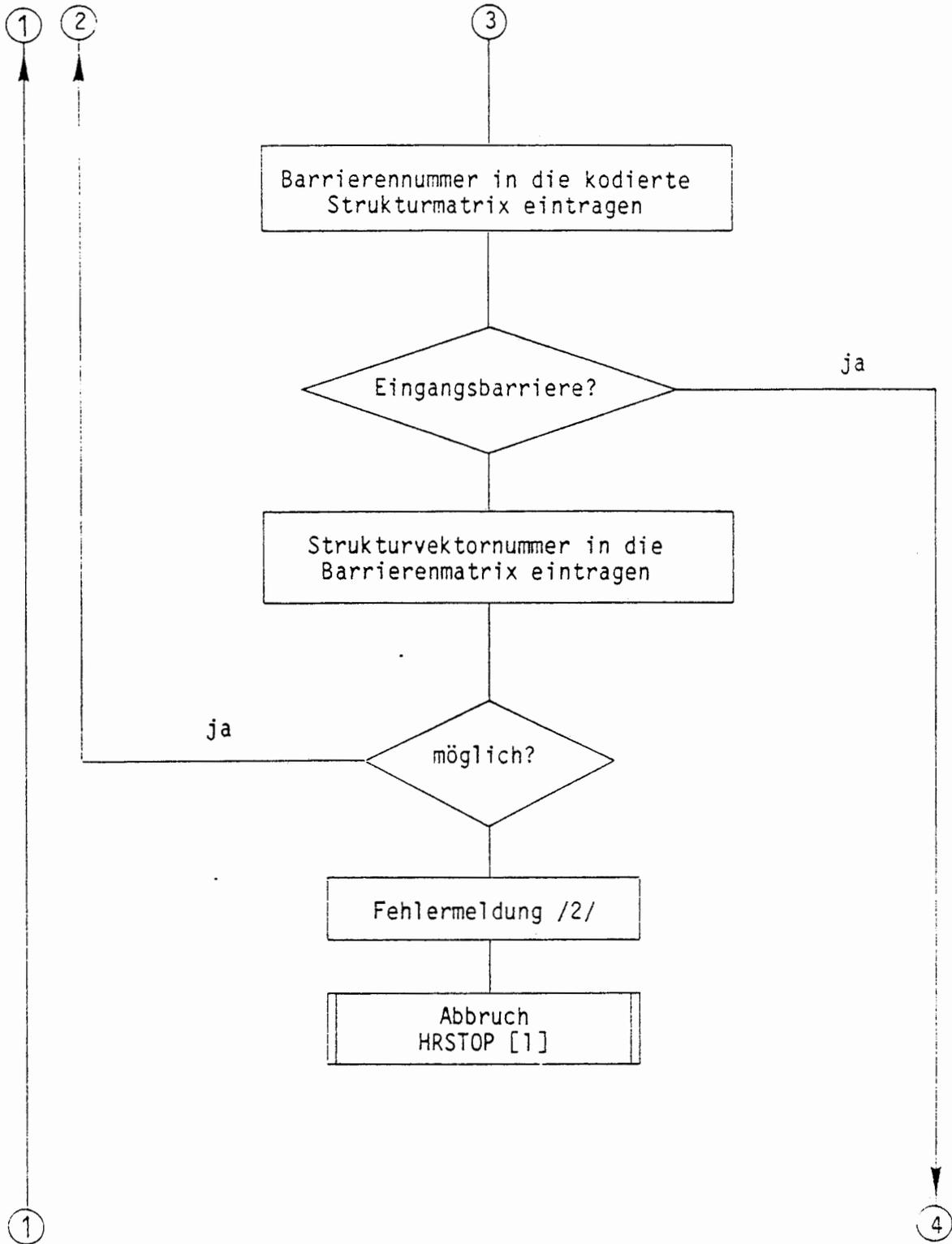
/WZSY/ : NISY, ISY
/WZBY/ : IB1Y

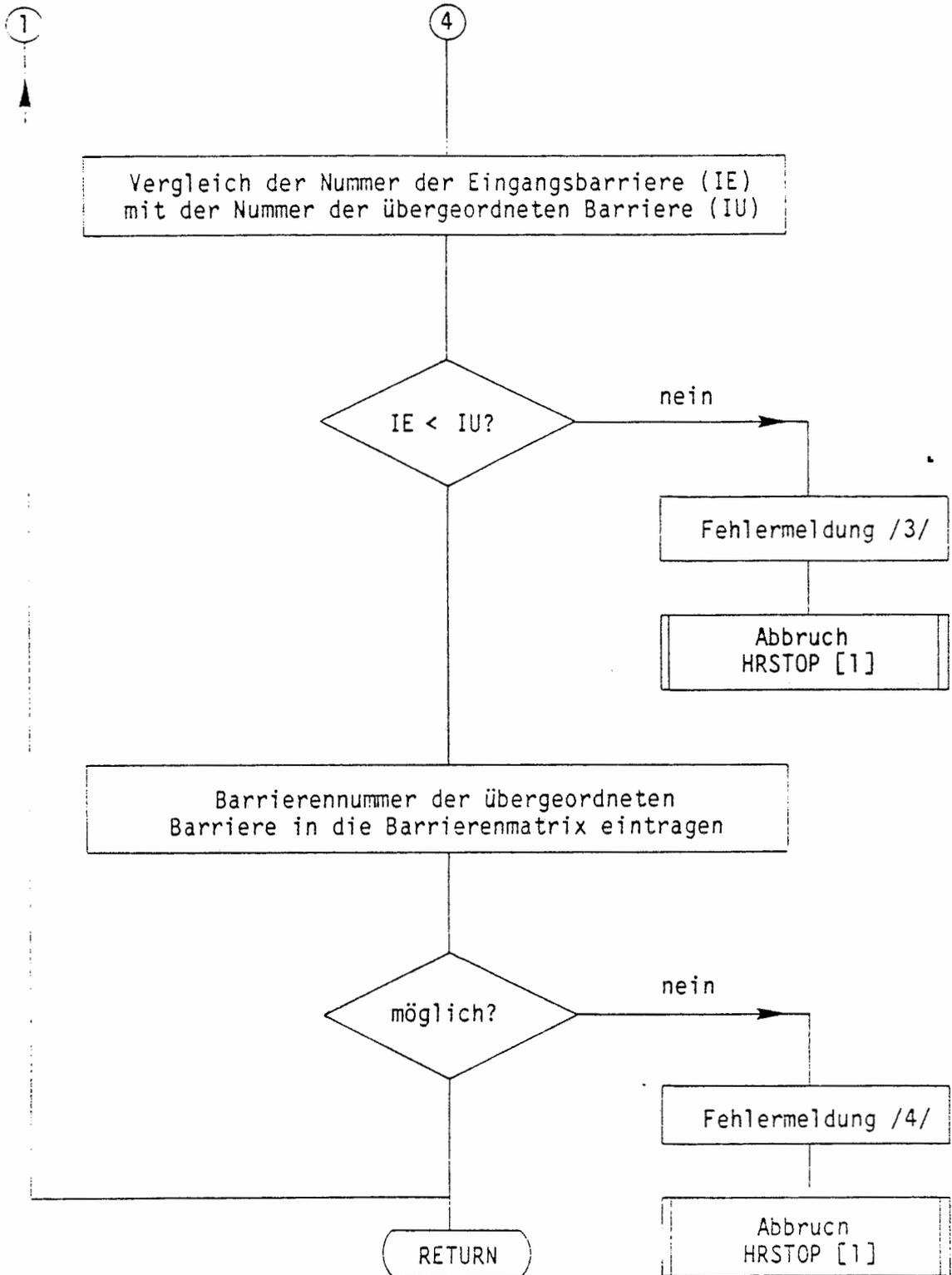
UNTERPROGRAMMAUFRUFE:

[1] HRSTOP('KSEBUE')

ABLAUF DES PROGRAMMS KSEBUE







A 2.4.6 ABFALLSPEZIFISCHE DATEN (ADATIN)

Zur Bereitstellung der Radionuklidinventare für Mobilisierungsmodelle werden Datensätze mit abfallspezifischen Daten von der Datenbibliothek übernommen. Ein abfallspezifischer Datensatz enthält die Aktivitätsinventare aller Nuklide sowie einige allgemeine Daten zur Charakterisierung eines Abfallgebundes. Der abfallspezifische Datensatz wird durch den Abfallnamen eindeutig bestimmt.

In den barrierenspezifischen Daten für ein Mobilisierungsmodell ist ein Name zur Charakterisierung des zugehörigen Abfalls enthalten. Für den Fall, daß dieser Name ein einzelnes Abfallgebunde charakterisiert, wird der zugehörige abfallspezifische Datensatz aus der Bibliothek übernommen und in die Abfallmatrix für die entsprechende Barriere eingetragen. Andernfalls charakterisiert der Name eine Mischung verschiedener Abfallgebunde. Mit Kenntnis der Zusammensetzung der Abfallmischung werden die zugehörigen abfallspezifischen Datensätze aus der Bibliothek übernommen, gewichtet aufsummiert und in die Abfallmatrix für die entsprechende Barriere eingetragen.

Bei der praktischen Übernahme der abfallspezifischen Daten wird eine mehr programmtechnisch orientierte Vorgehensweise eingehalten, bei der als erstes die Abfallmischungen eingelesen werden. Danach wird jeweils ein abfallspezifischer Datensatz von der Bibliothek übernommen und in einem Zwischenspeicher abgelegt (ABDAIN). Hinsichtlich des Namens des übernommenen Datensatzes werden dann die barrierenspezifischen Daten und die zugehörigen Abfallmischungen durchsucht. Für die erforderlichen Barrieren wird der abfallspezifische Datensatz entweder direkt in die Abfallmatrix eingetragen oder im Fall einer Mischung in der Abfallmatrix gewichtet aufsummiert. Nach der Abarbeitung aller abfallspezifischen Datensätze in der Bibliothek ist die Übertragung abfallspezifischer Daten in die Abfallmatrix abgeschlossen. Die Abfallmatrix /WZAY/ ist dann erstellt.

Die Aktivitätsinventare im abfallspezifischen Datensatz beziehen sich im allgemeinen auf den Zeitpunkt der Herstellung des Abfallgebundes. Da bis zum Zeitpunkt der Einlagerung ins Endlager oftmals eine größere

A 2.4.6 ABFALLSPEZIFISCHE DATEN (ADATIN)

Zur Bereitstellung der Radionuklidinventare für Mobilisierungsmodelle werden Datensätze mit abfallspezifischen Daten von der Datenbibliothek übernommen. Ein abfallspezifischer Datensatz enthält die Aktivitätsinventare aller Nuklide sowie einige allgemeine Daten zur Charakterisierung eines Abfallgebundes. Der abfallspezifische Datensatz wird durch den Abfallnamen eindeutig bestimmt.

In den barrierenspezifischen Daten für ein Mobilisierungsmodell ist ein Name zur Charakterisierung des zugehörigen Abfalls enthalten. Für den Fall, daß dieser Name ein einzelnes Abfallgebunde charakterisiert, wird der zugehörige abfallspezifische Datensatz aus der Bibliothek übernommen und in die Abfallmatrix für die entsprechende Barriere eingetragen. Andernfalls charakterisiert der Name eine Mischung verschiedener Abfallgebunde. Mit Kenntnis der Zusammensetzung der Abfallmischung werden die zugehörigen abfallspezifischen Datensätze aus der Bibliothek übernommen, gewichtet aufsummiert und in die Abfallmatrix für die entsprechende Barriere eingetragen.

Bei der praktischen Übernahme der abfallspezifischen Daten wird eine mehr programmtechnisch orientierte Vorgehensweise eingehalten, bei der als erstes die Abfallmischungen eingelesen werden. Danach wird jeweils ein abfallspezifischer Datensatz von der Bibliothek übernommen und in einem Zwischenspeicher abgelegt (ABDAIN). Hinsichtlich des Namens des übernommenen Datensatzes werden dann die barrierenspezifischen Daten und die zugehörigen Abfallmischungen durchsucht. Für die erforderlichen Barrieren wird der abfallspezifische Datensatz entweder direkt in die Abfallmatrix eingetragen oder im Fall einer Mischung in der Abfallmatrix gewichtet aufsummiert. Nach der Abarbeitung aller abfallspezifischen Datensätze in der Bibliothek ist die Übertragung abfallspezifischer Daten in die Abfallmatrix abgeschlossen. Die Abfallmatrix /WZAY/ ist dann erstellt.

Die Aktivitätsinventare im abfallspezifischen Datensatz beziehen sich im allgemeinen auf den Zeitpunkt der Herstellung des Abfallgebundes. Da bis zum Zeitpunkt der Einlagerung ins Endlager oftmals eine größere

Zeit vergeht, kann in den allgemeinen Daten zur Charakterisierung eines Abfallgebundes eine Zwischenlagerdauer eingegeben werden. Vor der Eintragung eines abfallspezifischen Datensatzes in die Abfallmatrix werden die Aktivitätsinventare entsprechend dem radioaktiven Zerfall in der Zwischenlagerzeit modifiziert (ZWLAG). Nach der Modifikation der Abfallinventare bezüglich des radioaktiven Zerfalls wird der abfallspezifische Datensatz vor der Eintragung in die Abfallmatrix auf Anforderung ausgedruckt (DRUABDA).

Während der Übernahme eines Datensatzes und insbesondere nach der Übertragung aller Datensätze in die Abfallmatrix werden verschiedene Fehlerkontrollen durchgeführt:

- Überschreitet die Anzahl der Mixturen die Dimensionierung des zugehörigen Speicherbereichs?
- Wird für eine Barriere sowohl die Übernahme eines Einzelgebundes als auch die Übernahme für eine Mischung gefordert?
- Wird ein abfallspezifischer Datensatz für eine Barriere angefordert, die kein Abfallgebunde ist?
- Ist der angegebene Name zur Charakterisierung eines Abfalls weder als Abfallgebunde noch als Mischung vorhanden?
- Ist das geforderte Abfallgebunde einer Mischung in der Bibliothek vorhanden?
- Stimmt die Zahl der Einzelabfälle für eine Abfallmischung in den barrierenspezifischen Daten mit derjenigen in der Mischung überein?
- Stimmt die Gesamtgebundenzahl für die Abfallmischung in den barrierenspezifischen Daten mit derjenigen in der Mischung überein?
- Stimmt die Gesamtgebundenzahl in der Mischung mit der Summe aller Einzelgebundenzahlen der Mischung überein?

ÜBERSICHT ÜBER DAS PROGRAMM ADATIN

ÜBERGEORDNETES PROGRAMM : DALEKO

LÄNGE DES PROGRAMMS : 425 Zeilen
1992 Bytes

STAND DER DOKUMENTATION : 15. März 1986

KURZBESCHREIBUNG:

Eingabe von Abfallmixturen vom Job-Input-File und von abfallspezifischen Datensätzen von der Bibliothek und Aufbau der Abfallmatrix

LESENDER ZUGRIFF AUF COMMON-BLÖCKE:

/RZ/ : IIN(3)
/WTEXT/ : CBLANK
/WCBX/, /WCAX/, /WZAX/
/WCBY/
/WZBY/ : NBY, IB1Y
/WZSY/ : NSY
/WZNY/ : NNY

SCHREIBENDER ZUGRIFF AUF COMMON-BLÖCKE:

/WZAY/

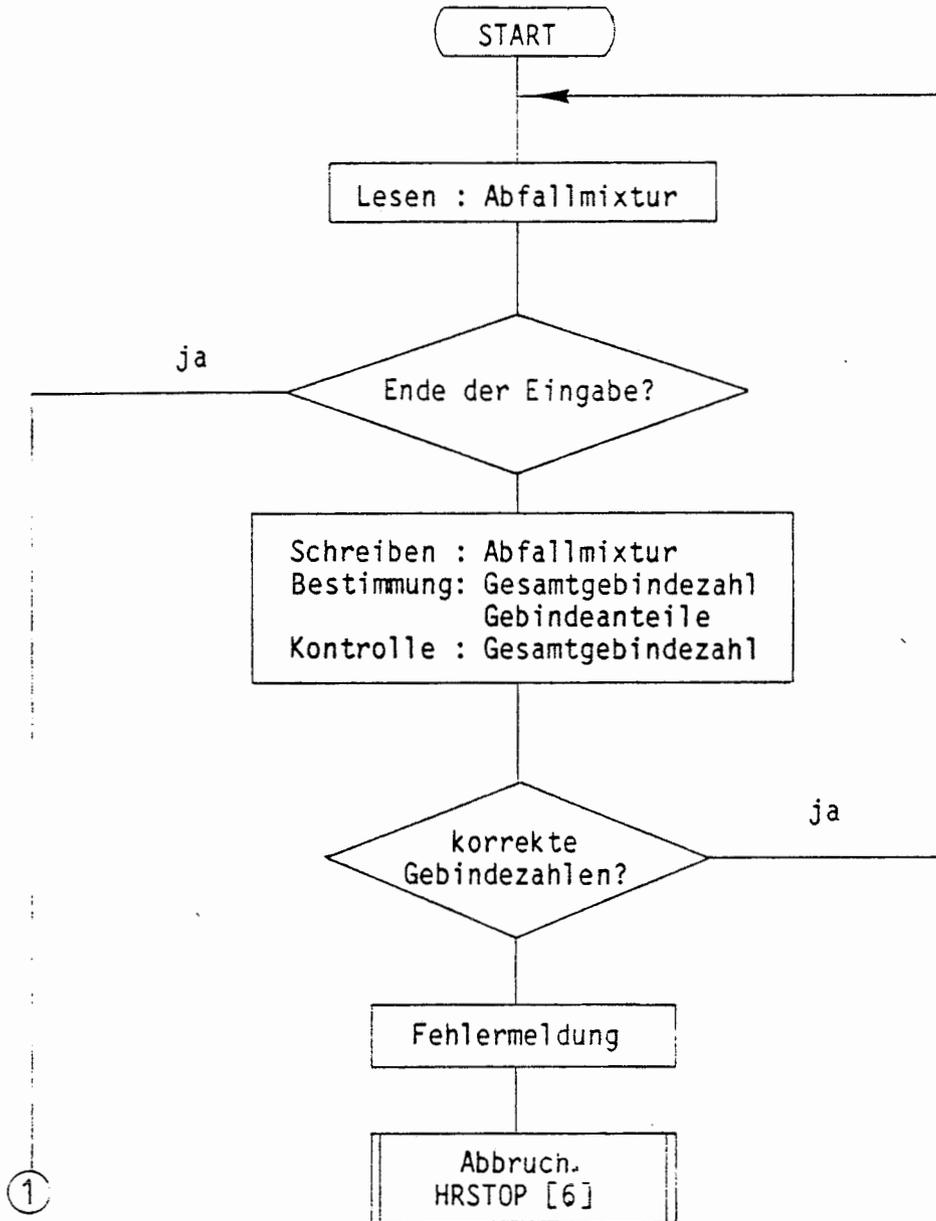
ZUGRIFF BEIM UNTERPROGRAMMAUFRUF:

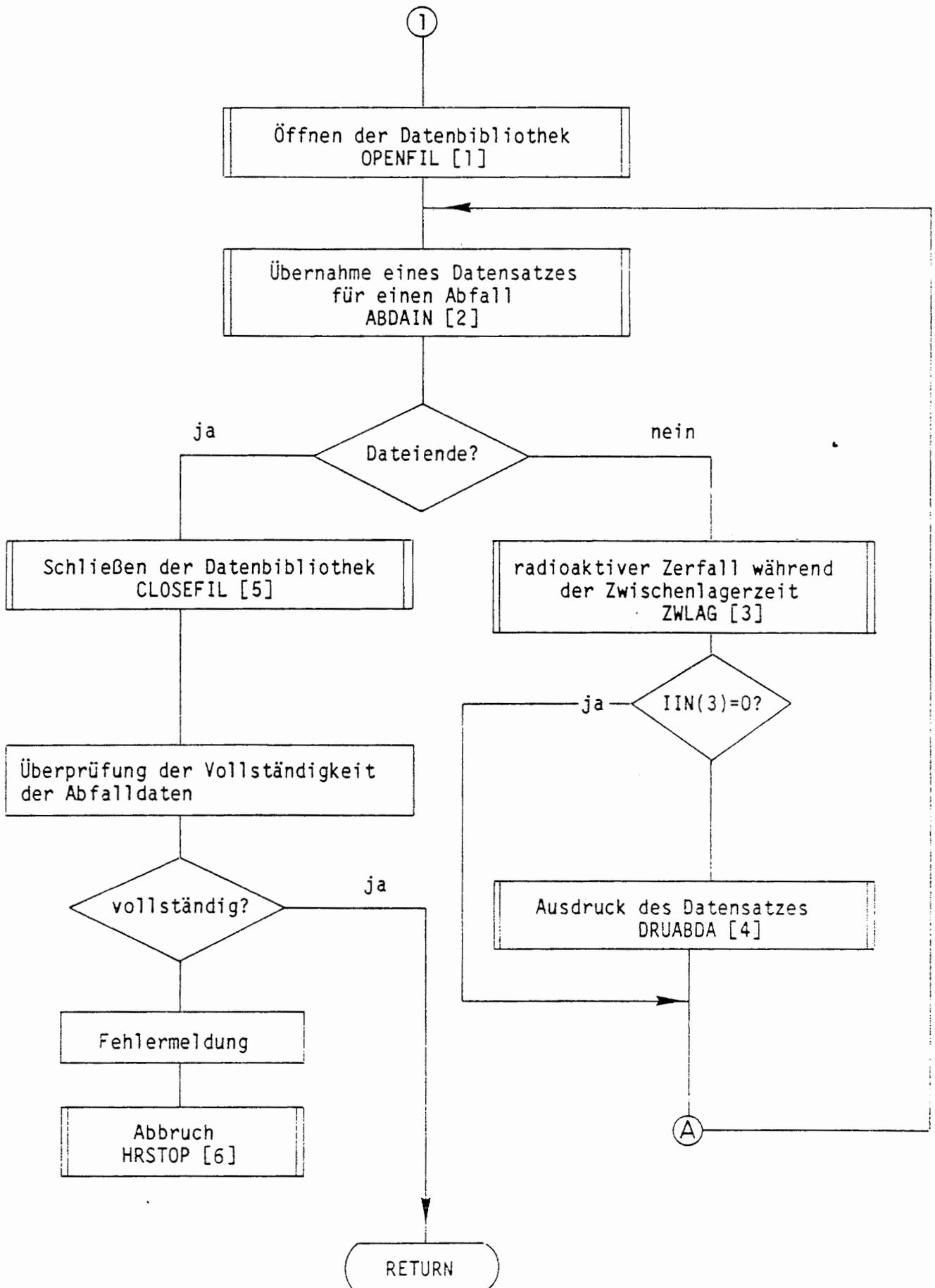
// : IST(3)
/WZAX/ : RAB1X
/WCAX/ : CAX
/WZNY/ : NNY

UNTERPROGRAMMAUFRUFE:

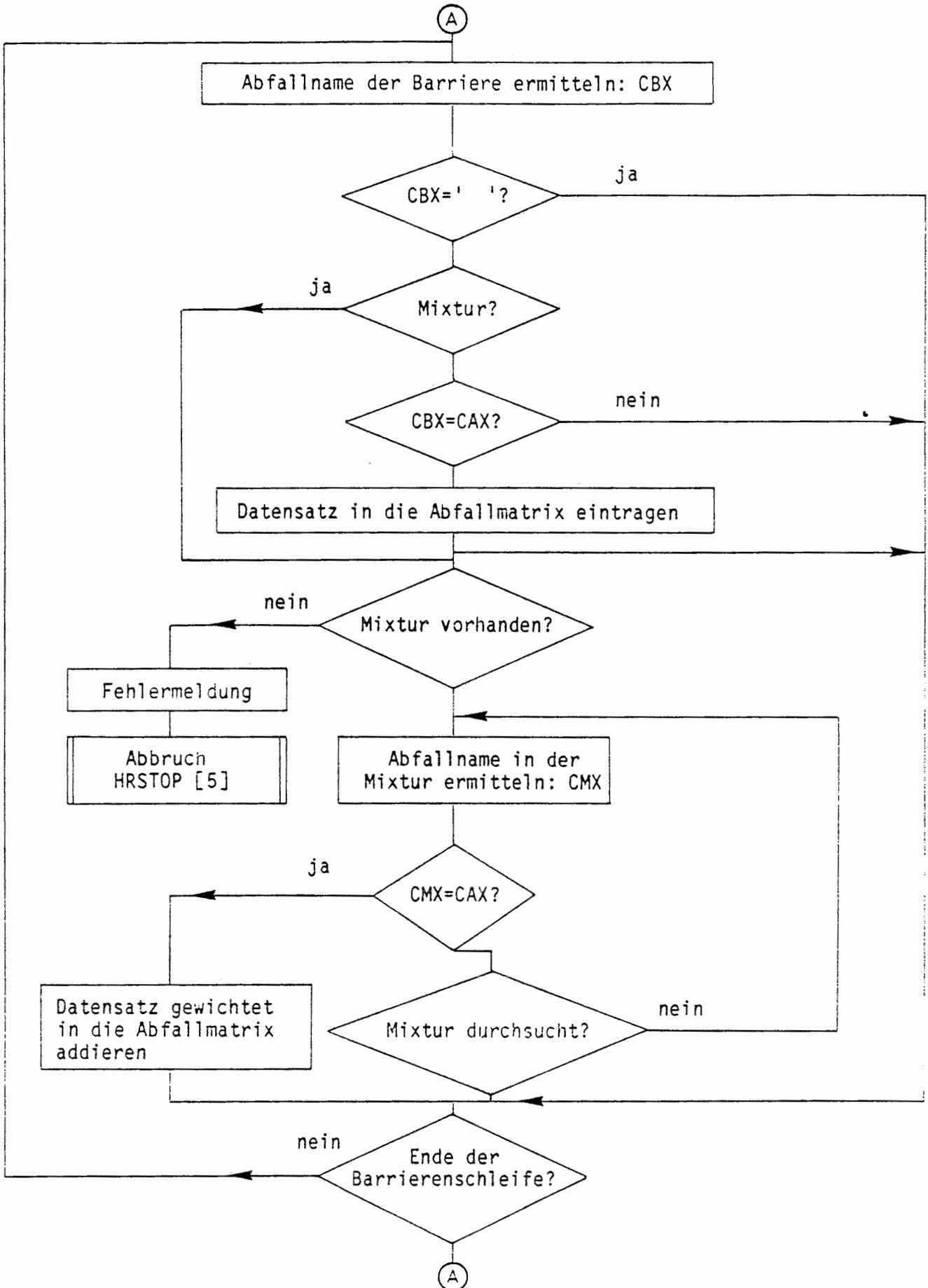
[1] OPENFIL (IST(3), 'ABDA')
[2] ABDAIN (DATEND)
[3] ZWLAG
[4] DRUABDA (RAB1X, NNY,-,1, CAX,1)
[5] CLOSEFIL ('ABDA')
[6] HRSTOP ('ADATIN')

ABLAUF DES PROGRAMMS ADATIN





ABLAUF DES PROGRAMMS ADATIN TEILAUFGABE A



A 2.4.6.1 EINGABE ABFALLSPEZIFISCHER DATEN (ABDAIN)

Bei der Übernahme abfallspezifischer Daten von der Bibliothek wird beim Aufruf von ABDAIN ein Datensatz mit Aktivitätsinventaren und allgemeinen Daten des Abfallgebundes sowie dem Namen des Abfallgebundes in Zwischenspeicherbereiche (/WCAX/, /WZAX/) geschrieben. Beim Erreichen des Dateiendes wird die Parametervariable DATEND='TRUE' gesetzt.

ÜBERSICHT ÜBER DAS PROGRAMM ABDAIN

ÜBERGEORDNETES PROGRAMM : DALEKO

LÄNGE DES PROGRAMMS : 97 Zeilen
 267 Bytes

STAND DER DOKUMENTATION : 15. März 1986

KURZBESCHREIBUNG:

Einlesen abfallspezifischer Daten für ein Gebinde aus der Bibliothek

PARAMETERLISTE:

DATEND

AUSGANGSPARAMETER:

DATEND : Kennung für Dateiende

LESENDER ZUGRIFF AUF COMMON-BLÖCKE:

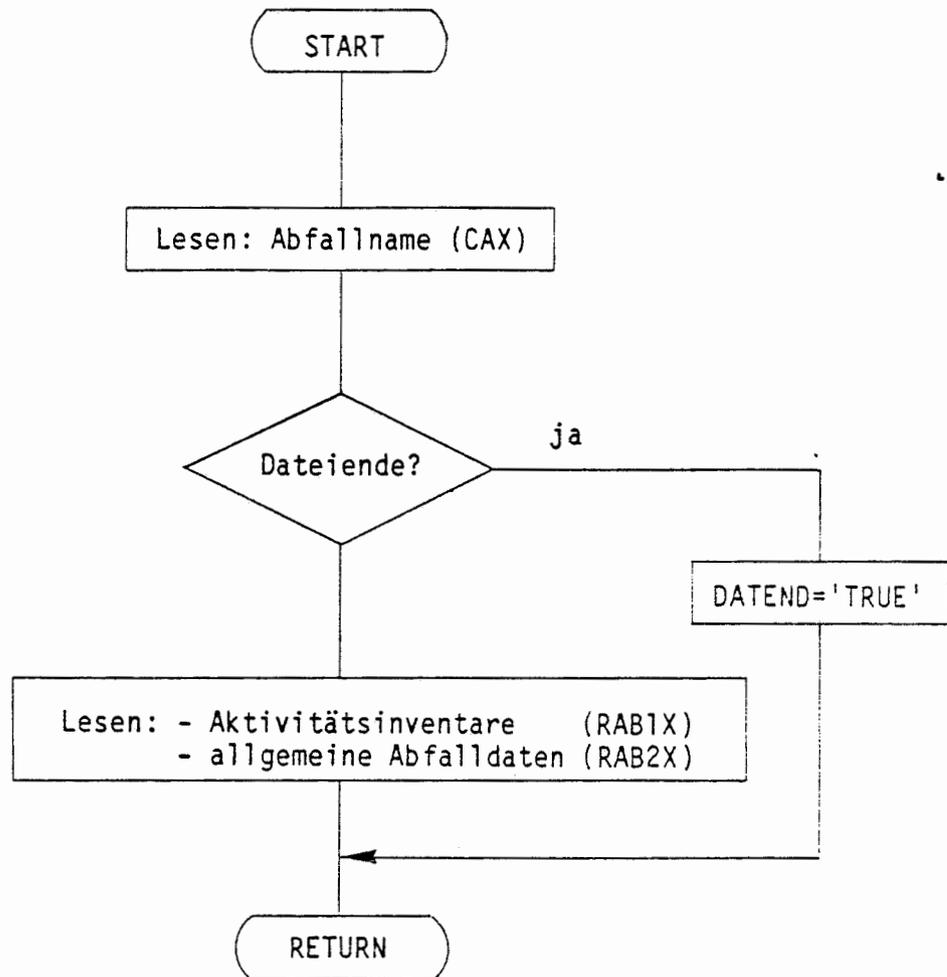
/WZNY/ : NNY

SCHREIBENDER ZUGRIFF AUF COMMON-BLÖCKE:

/WCAX/

/WZAX/

ABLAUF DES PROGRAMMS ABDAIN



A 2.4.6.2 ZWISCHENLAGERUNG DER ABFALLGEBINDE (ZWLAG)

Die Aktivitätsinventare in einem abfallspezifischen Datensatz beziehen sich im allgemeinen auf den Zeitpunkt der Herstellung des Abfallgebundes. Vom Zeitpunkt der Herstellung bis zum Zeitpunkt der Einbringung ins Endlager werden die Gebinde zwischengelagert. Zur Berücksichtigung der Zwischenlagerzeit berechnet das Programm ZWLAG die veränderten Aktivitätsinventare eines Abfallgebundes für den Zeitpunkt der Einlagerung ins Endlager.

Die Berechnung des radioaktiven Zerfalls erfolgt unter Verwendung des Programms RAZE, was nach Unterteilung der Zwischenlagerdauer in 50 Zeitschritte bei jedem Zeitschritt aufgerufen wird.

ÜBERSICHT ÜBER DAS PROGRAMM ZWLAG

ÜBERGEORDNETES PROGRAMM : DALEKO
LÄNGE DES PROGRAMMS : 79 Zeilen
63 Bytes
STAND DER DOKUMENTATION : 15. März 1986

KURZBESCHREIBUNG:

Berechnung des radioaktiven Zerfalls während der Zwischenlagerzeit.

LESENDER ZUGRIFF AUF COMMON-BLÖCKE:

/WZAX/ : RAB2X(5)

SCHREIBENDER ZUGRIFF AUF COMMON-BLÖCKE:

/YZ/ : DT

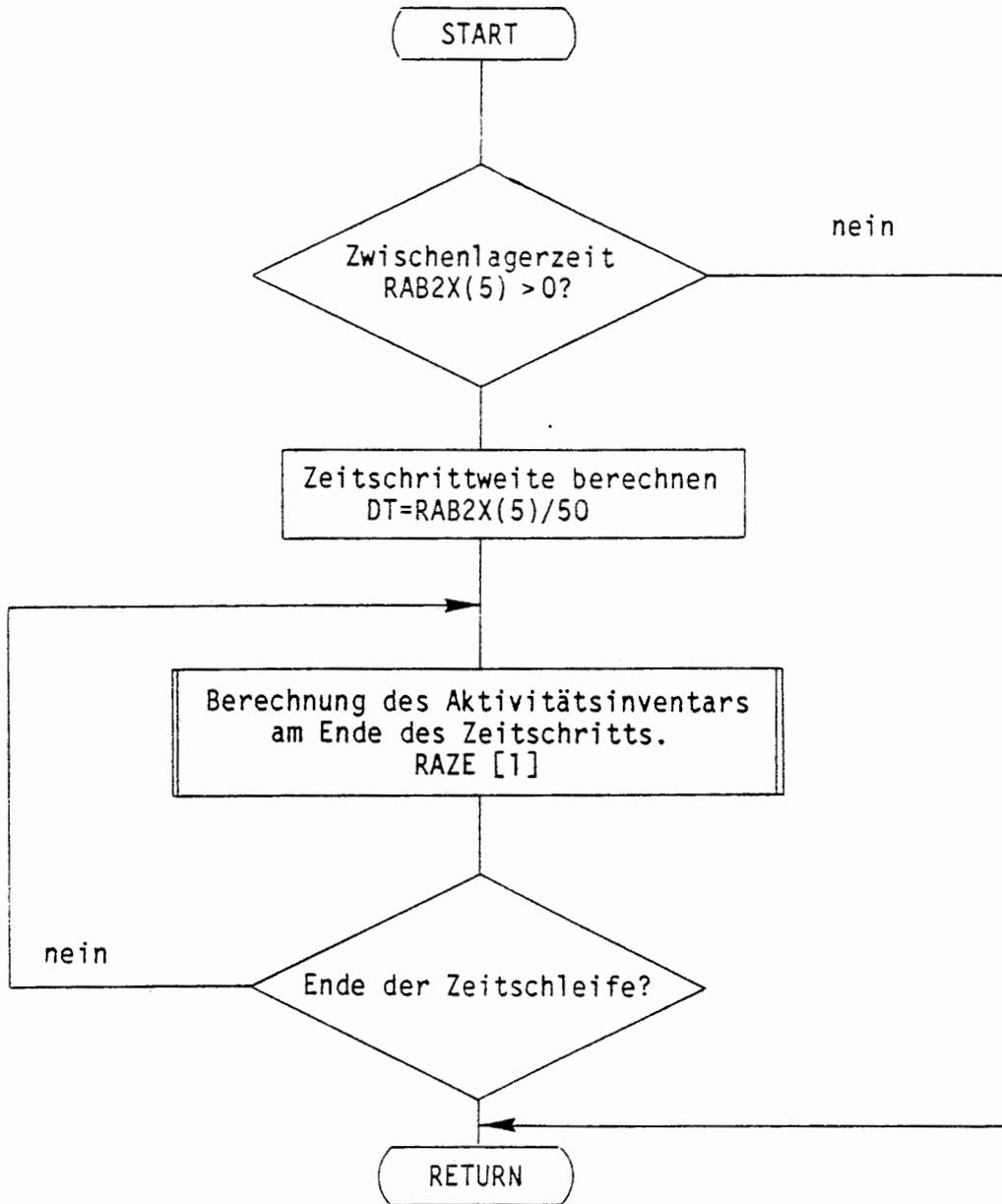
ZUGRIFF BEIM UNTERPROGRAMMAUFRUF:

RAB1X

UNTERPROGRAMMAUFRUFE:

[1] RAZE (0, RAB1X, RAB1X, RAB1X)

ABLAUF DES PROGRAMMS ZWLAG



A 2.4.6.3 AUSGABE ABFALLSPEZIFISCHER DATEN (DRUABDA)

Während der Eingabe abfallspezifischer Daten aus der Bibliothek besteht durch Aufruf des Programms DRUABDA die Möglichkeit der Ausgabe eines abfallspezifischen Datensatzes. Da ähnliche Anforderungen auch bei der Durchführung der zeitdiskreten Rechnung entstehen, enthält das Programm DRUABDA einen Schalter, der auf die Unterschiede bei der Ausgabe an verschiedenen Stellen des Programmablaufs eingeht.

Bei allen Schalterstellungen werden Zahlenwerte für alle betrachteten Nuklide geordnet nach Spaltprodukten und den vier Zerfallsreihen ausgedruckt. Die Zahlenwerte können je nach Schalterstellung und Eingabe eines entsprechenden Datenvektors über die Parameterliste (RDDAT) unterschiedlich sein. Folgende Schalterstellungen sind möglich:

IP = 0:

- Überschrift: Abfallspezifische Daten eines Gebindes
- Nuklidauswahl falls $IAU(2) > 0$
- Ausdruck mit Nuklidzuordnung

/IP/=1:

- Überschrift: Nuklidinventar in Bq einer Barriere falls $IP = -1$
- Nuklidauswahl falls $IAU(2) > 0$
- Ausdruck mit Nuklidzuordnung

/IP/=2:

- Überschrift: Nuklidinventar in kg einer Barriere falls $IP = -2$
- Nuklidauswahl falls $IAU(2) > 0$
- Umrechnung von Bq nach kg mit RNDAT
- Ausdruck mit Nuklidzuordnung

/IP/=3:

- Nuklidauswahl falls $IAU(2) > 0$
- Umrechnung in relative Größen durch Division der Größen durch RNDAT
- Ausdruck mit Nuklidzuordnung

/IP/=4:

- Berechnung der Wärmeleistung aus den eingegebenen Aktivitäten (RDDAT) unter Verwendung der Wärmekonversionsfaktoren
- Berechnung des Gefährdungspotentials aus den eingegebenen Aktivitäten (RDDAT) unter Verwendung der Dosiskonversionsfaktoren
- Berechnung der Aktinidenmassen für alle Aktinidenelemente aus den eingegebenen Aktivitäten (RDDAT) unter Verwendung der Massenkonzentrationsfaktoren.
- Ausdruck von Wärmeleistung, Gefährdungspotential und Aktinidenmassen.

/IP/=5:

- Berechnung der Massensummen von Elementen in den vier Zerfallsreihen aus den eingegebenen Aktivitäten (RDDAT) unter Verwendung der Massenkonzentrationsfaktoren. Die Elemente der Zerfallsreihen sind in COMMON /RC/ festgelegt.
- Ausdruck der Massensummen für die vier Zerfallsreihen.

/IP/=6:

- Ausdruck von 5 Werten der allgemeinen Gebindedaten aus RDDAT.

ÜBERSICHT ÜBER DAS PROGRAMM DRUABDA

ÜBERGEORDNETES PROGRAMM : REPOS

LÄNGE DES PROGRAMMS : 291 Zeilen
2426 Bytes

STAND DER DOKUMENTATION : 15. März 1986

KURZBESCHREIBUNG:

Berechnung und Ausdruck von Nuklidangaben und zugehörige Daten, für verschiedene Fälle, die durch den Schalter IP gesteuert werden.

PARAMETERLISTE:

RDDAT, (NRDDAT), RNDAT, (NRNDAT), NRNDAT, CNAME (10), IP

EINGANGSPARAMETER:

RDDAT : Eingangsdaten aus denen je nach Schalter IP zu druckende
Daten berechnet werden

CNAME : Abfall- oder Barrierenname

IP : Schalter für Ausdrucksteuerung

LESENDER ZUGRIFF AUF COMMON-BLÖCKE:

/RZ/ IAU (2), IOU(7),, IOU(10), ITHR, INPR, IUR, IAMR,
/WZNY/ NNY, RRY(3 KNY), RNY(4, KNY), RNY(5, KNY)
/WCNY) NNRY, CNY(1 KNY)
/RC/ CNTHR, CNNPR, CNUR, CNAMR

ABLAUF DES PROGRAMMS:

Da der Ablauf des Programms bei den jeweiligen Schalterstellungen im wesentlichen den Tätigkeiten bei den genannten Schalterstellungen entspricht, wird auf die Darstellung von Ablaufdiagrammen verzichtet.

A 2.4.7 INVENTAR DES BARRIERENSYSTEMS (BARINV)

Nach dem Ablauf des Programms ADATIN sind denjenigen Barrieren, die ein Abfallgebinde darstellen, Aktivitätsinventare zugewiesen und in der Abfallmatrix RA1Y abgelegt. Unter Verwendung der relativen Häufigkeiten einer jeden Barriere in den barrierenspezifischen Daten und den Verknüpfungsvorschriften in der Strukturmatrix lassen sich auch für die anderen Barrieren Aktivitätsinventare berechnen. Die Inventare der anderen Barrieren werden dabei so berechnet, daß sich das Gesamtinventar aller darunterstehenden Barrieren, d.h. das Gesamtinventar aller Abfallgebinde ergibt, die in der Barrierenstruktur diesem Teilbereich des Grubengebäudes zugeordnet sind.

Das Programm BARINV ermittelt die Aktivitätsinventare aller Barrieren und speichert sie in der Inventarmatrix RIY ab. Als erstes werden die Inventare derjenigen Barrieren, die ein Abfallgebinde darstellen, aus der Abfallmatrix RA1Y übernommen. Für die weiteren Barrieren werden die Inventare aus denjenigen der Eingangsbarrieren und deren relativen Häufigkeiten ermittelt. Nach dem Durchlaufen des gesamten Barrierensystems enthält die zuletzt bearbeitete Barriere das Inventar des gesamten Endlagers.

Parallel zur Bestimmung der Aktivitätsinventare weiterer Barrieren werden die allgemeinen abfallspezifischen Daten nach der gleichen Berechnungsvorschrift auch für die weiteren Barrieren bestimmt. Die ermittelten Werte werden in die entsprechenden Positionen der Abfallmatrix RA2Y eingetragen.

ÜBERSICHT ÜBER DAS PROGRAMM BARINV

ÜBERGEORDNETES PROGRAMM : DALEKO

LÄNGE DES PROGRAMMS : 138 Zeilen
384 Bytes

STAND DER DOKUMENTATION : 15. März 1986

KURZBESCHREIBUNG:

Berechnung des Inventars und anderer Abfalldaten für das gesamte
Barriersystem

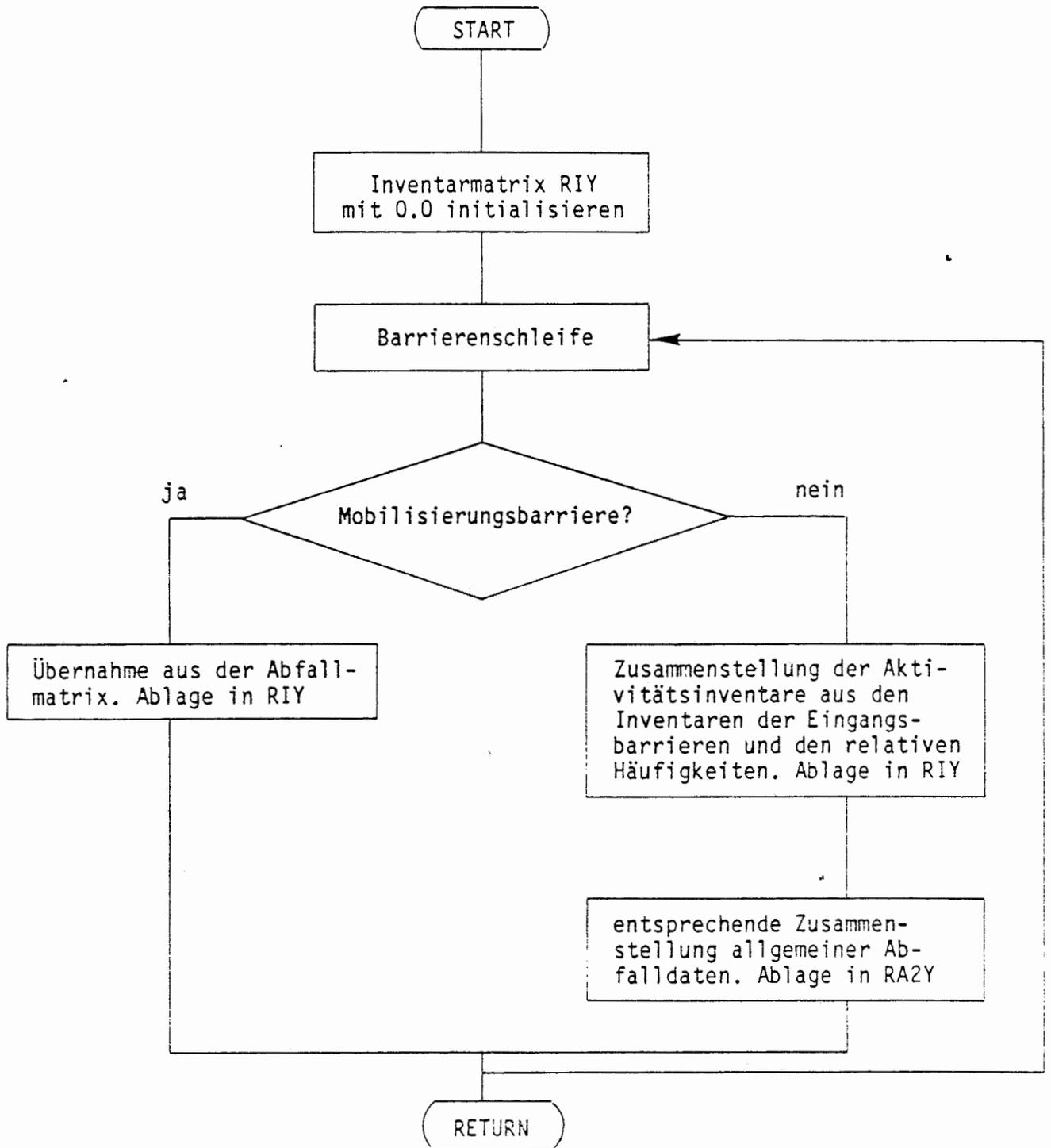
LESENDER ZUGRIFF AUF COMMON-BLÖCKE:

/WZII/ : RIY
/WZSY/ : ISY, NISY
/WTEXT/ : CBLANK
/WZBY/ : NBY, IB1Y
/WZNY/ : NNY
/WCBX/ : CBX
/WZAY/ : RA1Y, RA2Y

SCHRÉIBENDER ZUGRIFF AUF COMMON-BLÖCKE:

/WZII/ : RIY
/WZSY/ : NISY
/WZAY/ : RA2Y

ABLAUF DES PROGRAMMS BARINV



A 2.5 BERECHNUNG DER FREISETZUNG

Die numerische Auswertung des Endlagermodells erfolgt zeitdiskret im Unterprogramm STEBA. Innerhalb eines jeden Zeitschritts wird das gesamte Barrierensystem ausgewertet. Insbesondere werden die Ausgangsströme jeder Barriere aus den Eingangsströmen und den Barrieren-internen Vorgängen berechnet.

Zur Optimierung der Rechenzeit und der Rechengenauigkeit wird die Zeitschrittweite variabel gehalten. Eine geeignete Schrittweite für den jeweils nächsten Zeitschritt wird mit Hilfe von Kriterien ermittelt, die im Unterkapitel A 2.5.1 dargestellt werden.

Hauptaufgaben des Unterprogramms STEBA sind somit: Steuerung der Zeitschrittweite, Aufruf der Barrierenmodelle entsprechend dem vorgegebenen Barrierensystem, Sicherstellung des Datentransfers zwischen den Barrierenmodellen sowie Ausgabe der Ergebnisse.

Ausgegebene Größen sind z.B. Nuklid- und Lösungsströme, Freisetzungsmengen und -raten für ausgewählte Barrieren und Nuklide, Zeitpunkte des erstmaligen Erreichens der Löslichkeitsgrenzen der einzelnen Elemente und maximale Elementkonzentrationen. Alle diese Ausgaben werden über Parameter im Job-Input-File gesteuert.

Barrierenmodelle werden entsprechend dem aktuellen Wissensstand und den konkreten Anforderungen als Unterprogramme zur Verfügung gestellt. Innerhalb eines jeden Zeitschritts wird für jede Barriere das zugehörige Barrierenmodell aufgerufen (Unterprogramm BMOD, Kapitel A 2.5.2). Das Barrierenmodell selbst übernimmt den Aufruf und die Verknüpfung der Unterprogramme, in denen die betrachteten Einzeleffekte modelliert sind.

Die Reihenfolge, in der die einzelnen Barrierenmodelle abgearbeitet werden, richtet sich nach der Ausbreitungsrichtung der Radionuklide im Barrierensystem. Zur Berechnung der Ausgangsströme aus einer Barriere müssen die Eingangsströme in die Barriere bekannt sein. Daher werden in jedem Zeitschritt die innenliegenden vor den außenliegenden Barrieren abgearbeitet.

Diese Reihenfolge der Abarbeitung wird auch eingehalten zu Zeiten, zu denen noch keine Nuklid Ausbreitung erfolgt und die entsprechenden Einlagerungsorte noch nicht mit Lösung geflutet sind.

Die Auswertung des Barrierensystems beginnt also immer bei den Mobilisierungsmodellen, in denen die Aktivitätsströme aus den Abfallgebänden berechnet werden. Nach Multiplikation mit der Anzahl der Gebände im Einlagerungsort ist der Eingangsstrom für die nachfolgende Barriere (z.B. Bohrloch) bekannt, so daß die Freisetzungsströme berechnet werden können. Nach einem entsprechenden Übergang auf weitere Bereiche des Grubengebäudes in Abhängigkeit vom Endlagerdesign ergibt sich die Aktivitätsfreisetzung aus dem Endlager.

Aus dem unten angefügten Ablaufdiagramm wurden, um es übersichtlicher zu gestalten, einige Teilaufgaben (A bis H) herausgenommen. Diese Teilaufgaben stellen jeweils eine Programmeinheit dar, in der entweder Steuergrößen ermittelt werden (A, B, C, E, F) oder die Ausgabe von Ergebnissen vorbereitet wird (C) und Ausgaberroutinen angesprochen werden (D, G, H). Für die erste Gruppe werden am Ende des Teilbereiches im Ablaufdiagramm jeweils die ermittelten Größen aufgeführt. Die Ablaufdiagramme für die Teilaufgaben D bis H befinden sich in den zugehörigen Unterkapiteln.

ÜBERSICHT ÜBER DAS PROGRAMM STEBA

ÜBERGEORDNETES PROGRAMM : REPOS

LÄNGE DES PROGRAMMS : 737 Zeilen
7566 Bytes

STAND DER DOKUMENTATION : 15. März 1986

KURZBESCHREIBUNG:

Zeitschrittweise Abarbeitung des Barrierensystems, Aufruf des Unterprogramms BMOD zur Ansteuerung eines Barrierenmodells innerhalb einer Schleife über alle Barrieren in jedem Zeitschritt.

Ausgabe bzw. Speicherung von Rechenergebnissen für einen Zeitschritt innerhalb einer Barrierenschleife:

- Ausgabe von Nuklid- und Lösungsströmen vor und nach dem Aufruf von BMOD,
- Berechnung kumulierter Werte,
- Abspeicherung von Freisetzungsmengen,
- Ausgabe zeitabhängiger Größen für jeweils eine Barriere zu bestimmten Zeiten für ausgewählte Nuklide oder alle Nuklide, gesteuert durch Parameter im Job-Input-File.

Berechnung einer neuen Zeitschrittweite nach Verlassen der Barrierenschleife.

Ergebnisausgabe nach Ende des Szenarios:

- Ausgabe einer Tabelle über Zulaufzeiten, Erreichen der Endporosität bzw. Ende der Mobilisierung.
- Ausgabe von Löslichkeitstabellen, Tabellen mit maximalen Elementkonzentrationen,
- Ausdruck kumulierter Freisetzungsmengen,
- Ausdruck von Freisetzungsmengen und -raten jeweils für eine ausgewählte Barriere, gesteuert durch Parameter im Job-Input-File.

LESENDER ZUGRIFF AUF COMMON-BLÖCKE:

/YC/, /YZ/,
/RC/ : CNAU, CBAU
/RZ/,
/WCNY/, /WZNY/, /WCBY/, /WZBY/,
/WZSY/ : ISY
/WZAY/, /WZIY/, /WZUY/, /WZFY/,
/N/, /WTEXT/

SCHREIBENDER ZUGRIFF AUF COMMON-BLÖCKE:

/YC/,
/YZ/ : TA, TN, DT, DT1, DTA, DTMA, DTMI, NSTEP, IW, TNE, KB
/RZ/ : ZEFT
/WZFY/

ZUGRIFF BEIM UNTERPROGRAMMAUFRUF:

/YC/ : CBANA
/RZ/ : ZEFT
/WZNY/ : NNY
/WZBY/ : RB1Y, RB3Y, RB4Y, IB2Y
/WZAY/, /WZJY/, /WZUY/,
/WTEXT/ : CBLANK

UNTERPROGRAMMAUFRUFE:

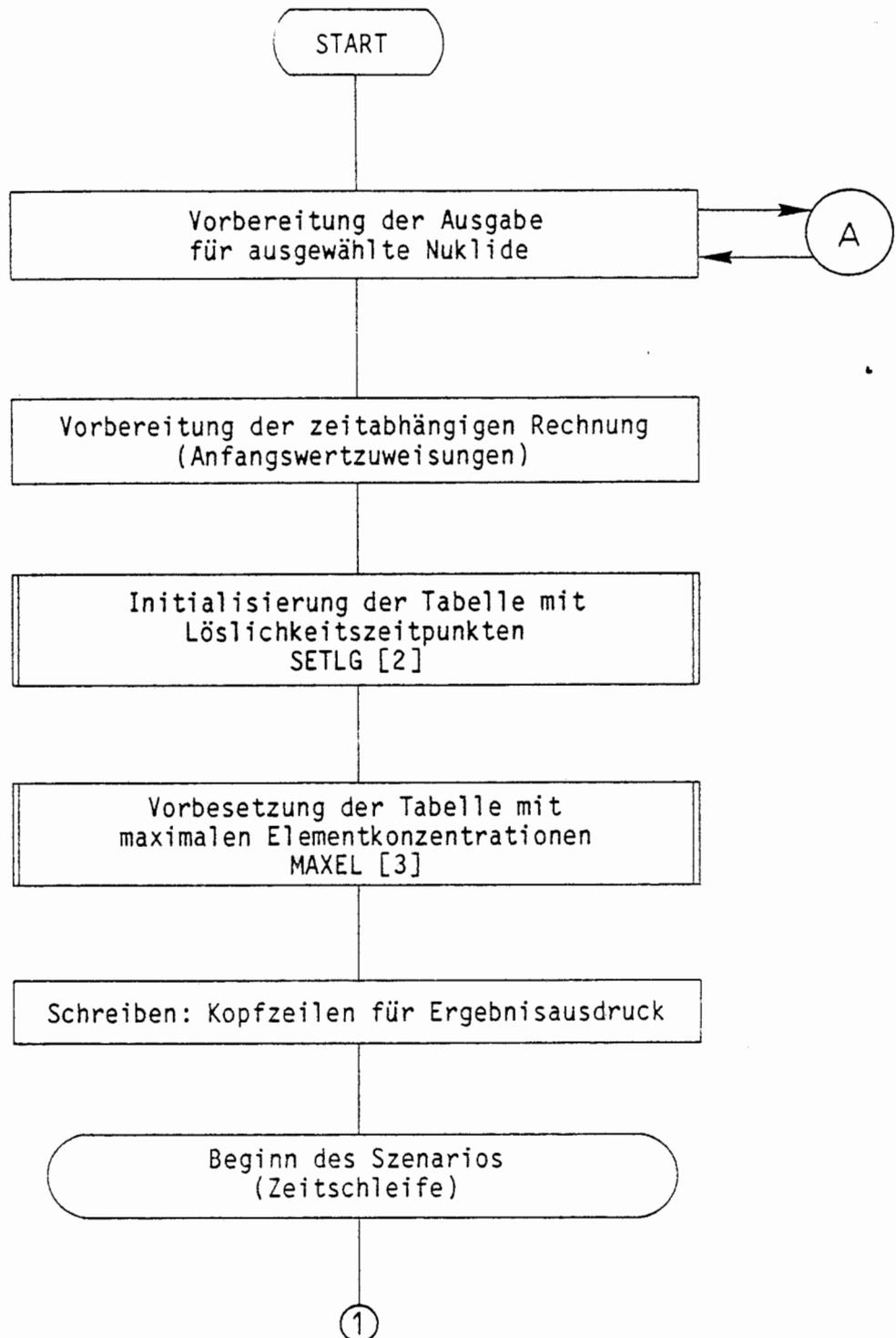
- [1] HRSTOP(CNAM)
- [2] SETLG (0)
- [3] MAXEL (0,0.,0.,0.,0.,0.,0.)
- [4] Ohne Bedeutung
- [5] BMOD (CMOD, EVN(1), EVS(1), IB2Y(1,KBY), RB4Y(1,1KBY),
RB3Y(1,KBY), RB1Y(1,KBY), IB2Y(1,KBAY),
RB4Y(1,1,KBAY), RB3Y(1,KBAY), RA1Y(1,KBY),
RA2Y(1,KBY), RU1Y(1,KBY), RU2Y(1,KBY), RU3Y(1,KBY))

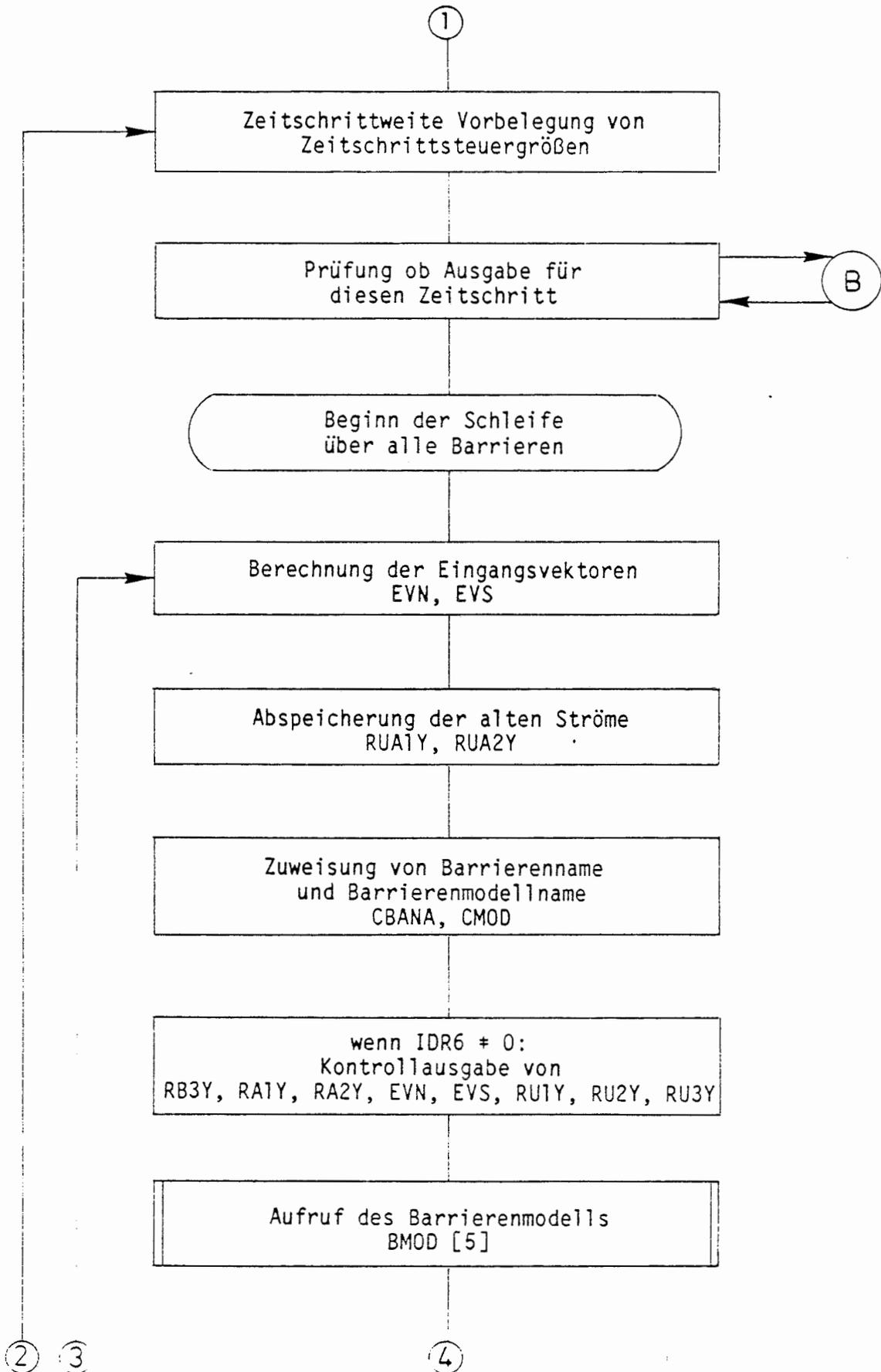
(KBY : Nummer der aktuellen Barriere,

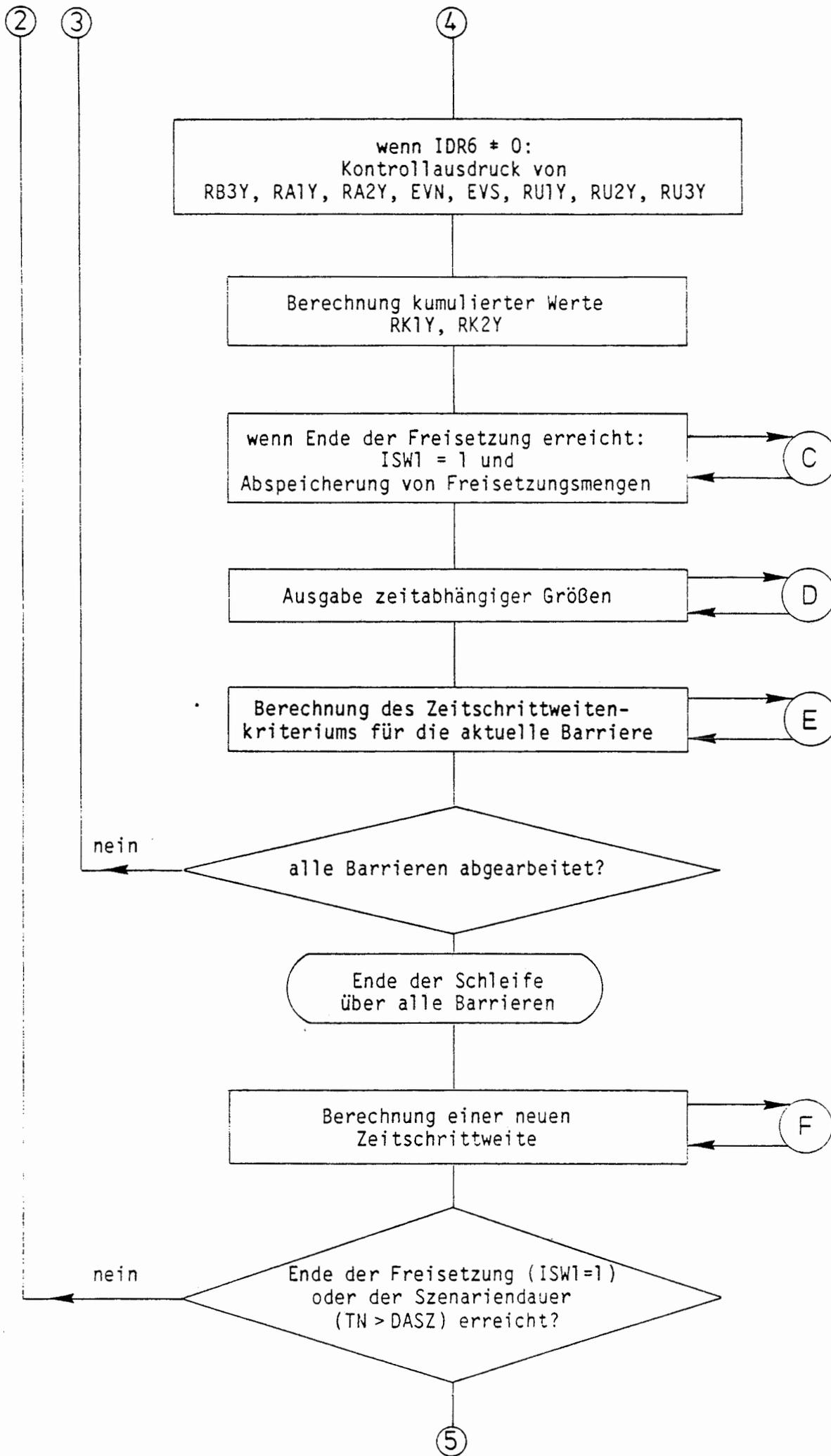
KBAY : Nummer der äußeren Barriere)

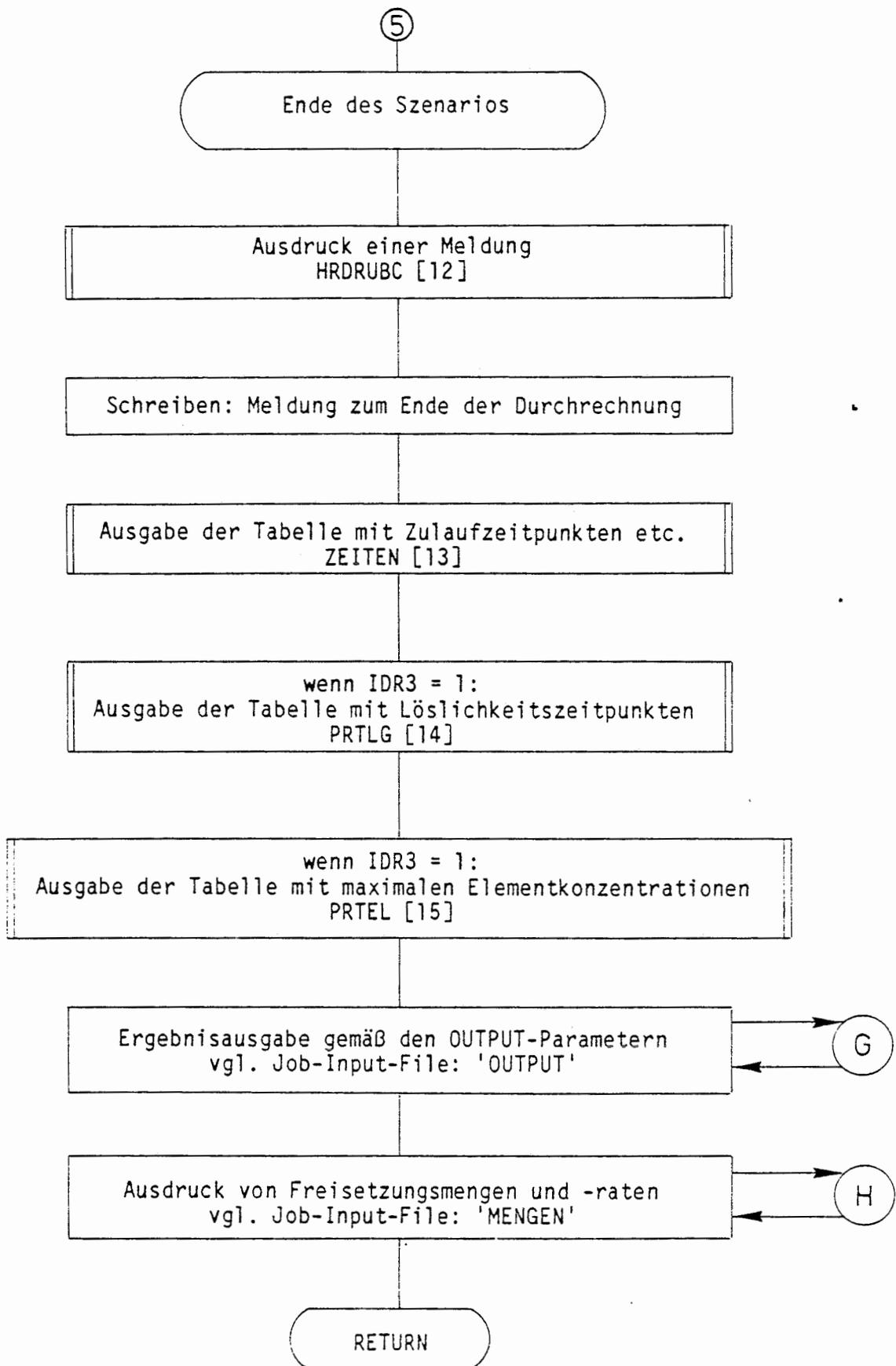
- [6] DRUABDA(RU1Y(1, KBY), NNY, DUMMY, 1, CBANA, 1)
- [7] DRUABDA(RK1Y(1, KBY), NNY, DUMMY, 1, CBLANK, 1)
- [8] DRUABDA(RA1Y(1, KBY), NNY, DUMMY, 1, C2 , 1)
- [9] DRUABDA(RB4Y(1,1,KBY), NNY, DUMMY, 1, C2 , 1)
- [10] DRUABDA(RB4Y(1,2,KBY), NNY, DUMMY, 1, C2 , 1)
- [11] DRUABDA(RB4Y(1,3,KBY), NNY, DUMMY, 1, C2 , 1)
- [12] HRDRUBC('SZENARIO DURCHGERECHNET')
- [13] ZEITEN
- [14] PRTLG
- [15] PRTEL
- [16] NOUTBAR(KBY)
- [17] DRUABDA(RK1Y(1,KBY), NNY, DUMMY, 1, C1 , 1)
- [18] DRUABDA(RK1Y(1,KBY), NNY, DUMMY, 1, C1 , 2)
- [19] DRUABDA(RK1Y(1,KBY), NNY, DUMMY, 1, CBLANK, 5)
- [20] DRUABDA(RK1Y(1,KBY), NNY, RIY(1,KBY), NNY,C1 , 3)
- [21] DRUFM(RKUM1, TFO, 20, CNA, ZEFT)
- [22] DRUFM(RKUM1, TFO, 21, CNA, ZEFT)
- [23] DRUFM(RKUM1, TFO, 23, CNA, ZEFT)
- [24] DRUFM(RKUM1, TFO, 25, CNA, ZEFT)
- [25] DRUFM(RKUM1, TFO, 26, CNA, ZEFT)
- [26] DRUFM(RKUM1, TFO, 27, CNA, ZEFT)
- [27] DRUFM(RKUM1, TFO, 28, CNA, ZEFT)

ABLAUF DES PROGRAMMS STEBA

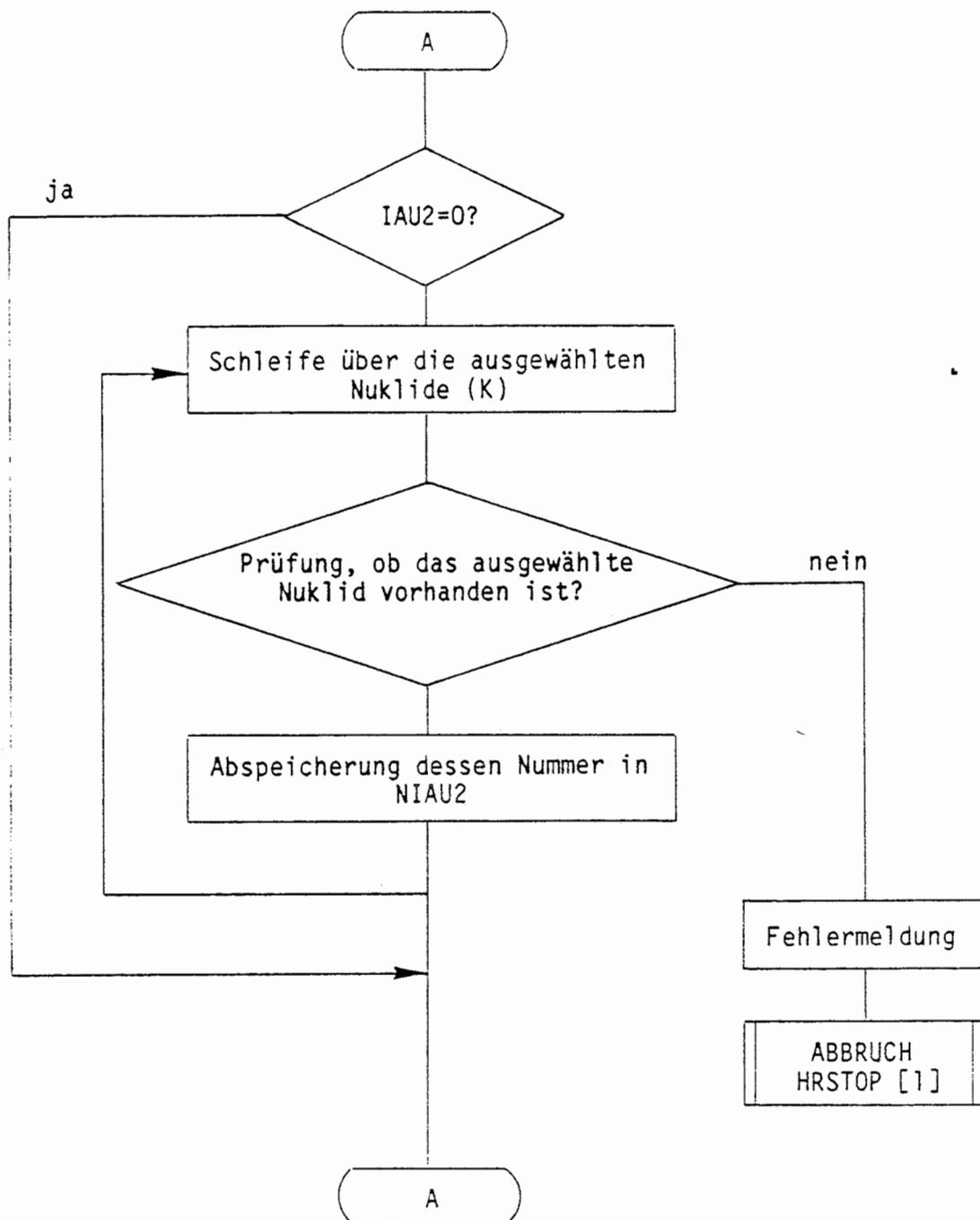








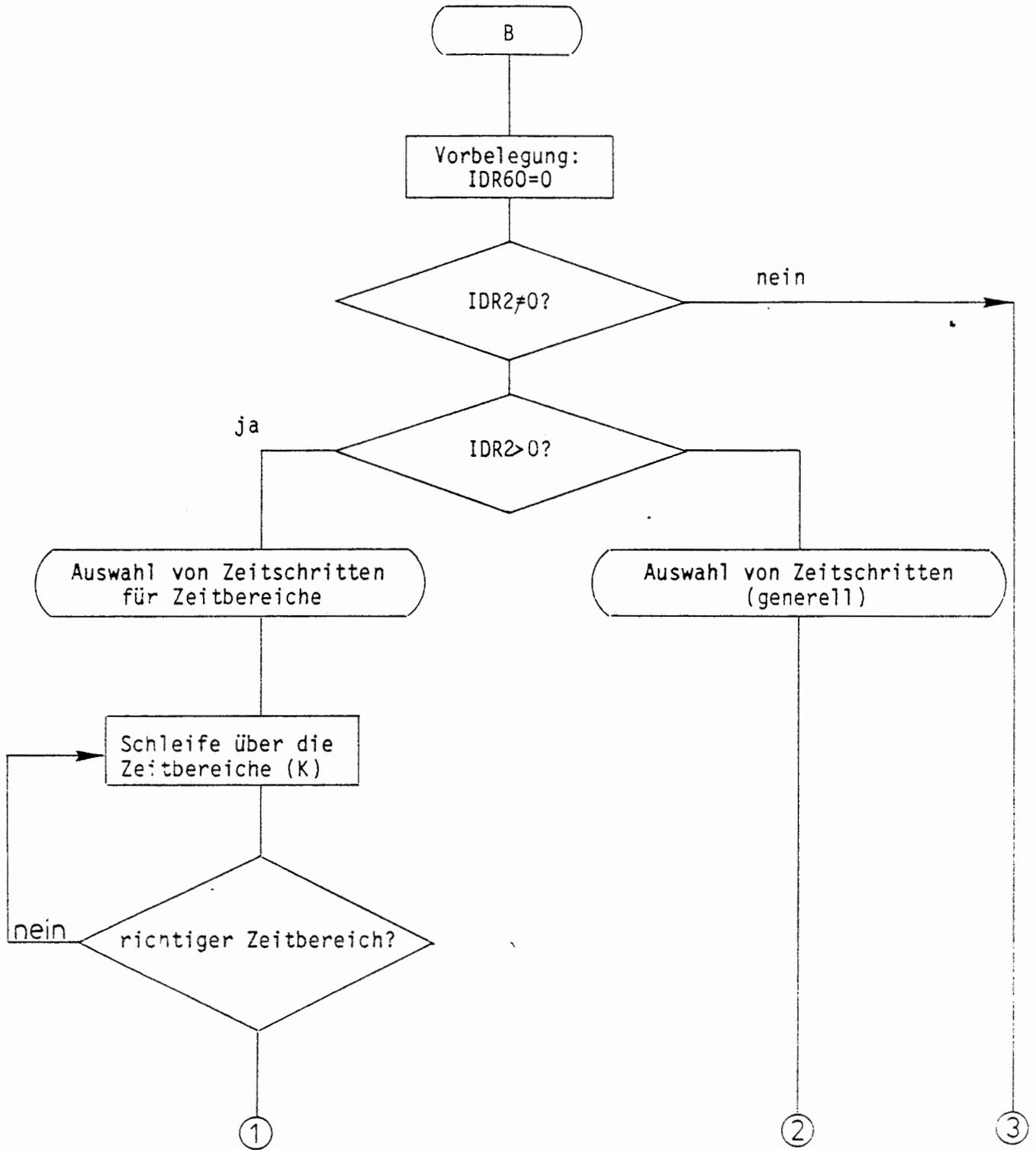
ABLAUF DES PROGRAMMS STEBA TEILAUFGABE A

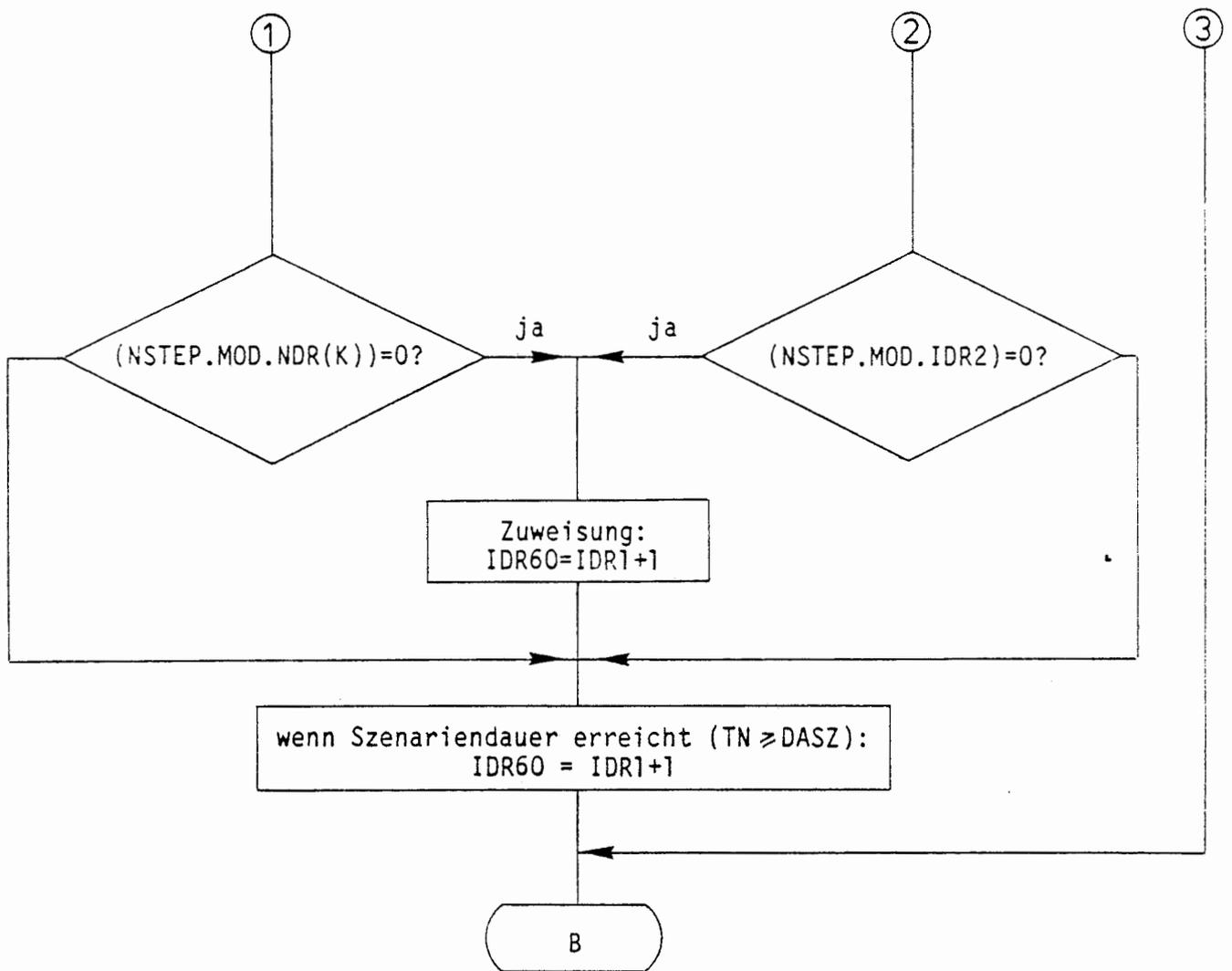


In dieser Teilaufgabe ermittelte Größen:

Nummern der zum Ausdruck ausgewählten Nuklide

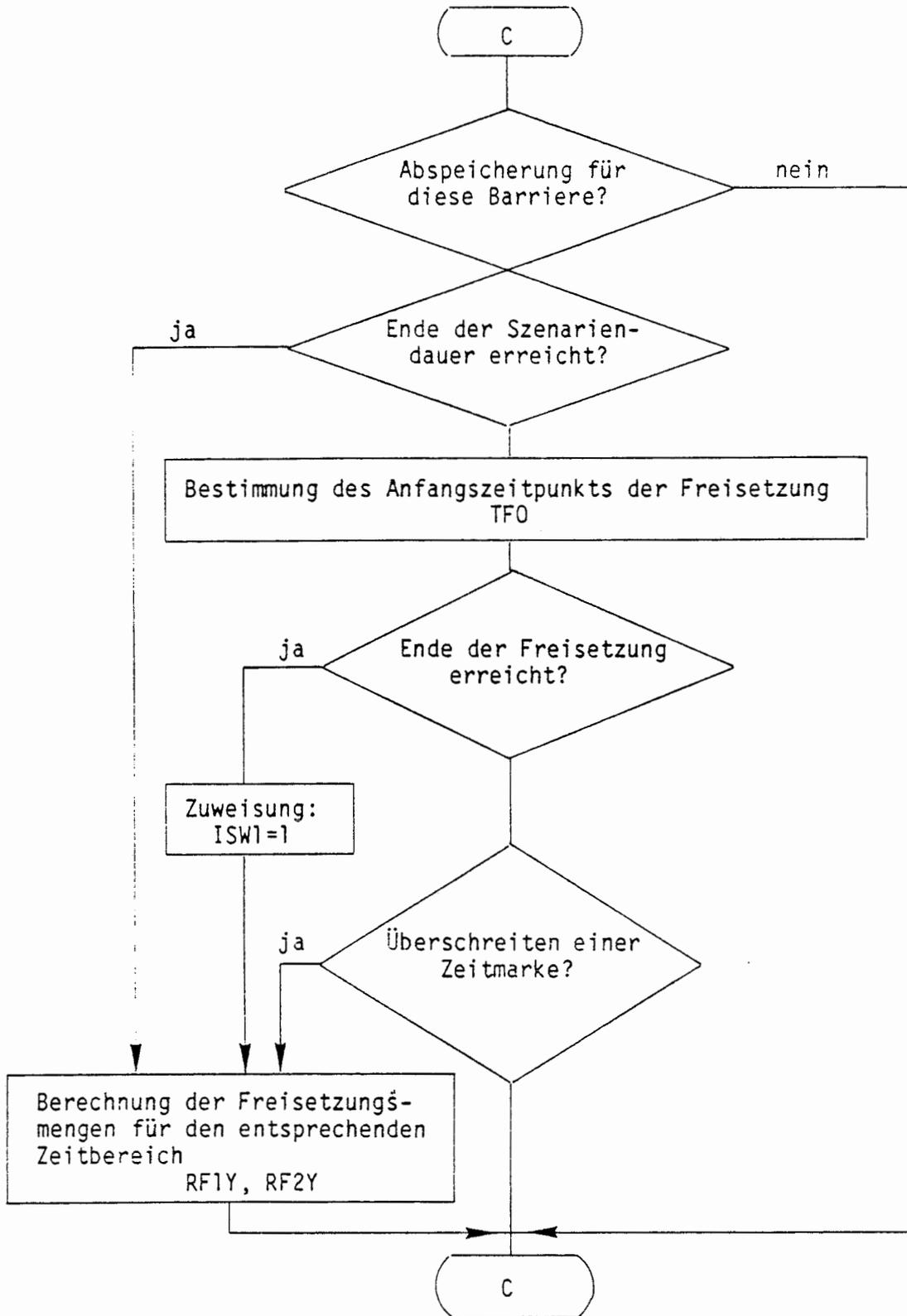
ABLAUF DES PROGRAMMS STEBA TEILAUFGABE B





In dieser Teilaufgabe ermittelte Größen:
Steuerparameter für die Ausdrucksteuerung (IDR60)

ABLAUF DES PROGRAMMS STEBA TEILAUFGABE C



In dieser Teilaufgabe ermittelte Größen:

Freigesetzte Mengen (RF1Y, RF2Y) und Steuerparameter, der anzeigt, ob die Freisetzung abgeschlossen ist (ISW1)

A 2.5.1 SCHRITTWEITENSTEUERUNG

Die zeitdiskrete Durchrechnung des Szenarienablaufs beginnt mit einer im Job-Input-File vorgesehenen Schrittweite DT1, die gleichzeitig die Mindestschrittweite darstellt. Diese Schrittweite wird für eine Anzahl von NSTEPM Zeitschritten (zur Zeit 20) beibehalten. Danach wird die Schrittweite nach folgendem Mechanismus gesteuert.

Nach Durchrechnung aller Barrieren für den aktuellen Zeitschritt TN wird vor dem Übergang zum nächsten Zeitpunkt die Zeitschrittweite DT überprüft und ggf. neu festgelegt.

Berechnung des Schrittweitenkriteriums (vgl. Teilaufgabe E im Ablaufdiagramm):

Bei der Überprüfung der Zeitschrittweite wird das Maximum der relativen Änderung der Aktivitätsströme bezüglich aller Nuklide und Barrieren ermittelt.

$$\text{EPS} = \text{MAX}_{(\text{Barrieren})} \left(\text{MAX} \left[\text{MAX}_{(\text{Nuklide})} \left| \frac{|RU1Y| - |RUA1Y|}{RUA1Y} \right|, \right. \right. \\ \left. \left. \text{MAX}_{(\text{Ströme})} \left| \frac{|RU2Y| - |RUA2Y|}{RUA2Y} \right| \right] \right),$$

wobei RU1Y die Nuklidübergangsströme zur aktuellen Zeit TN, RUA1Y die zur Zeit TN-DT sowie RU2Y die anderen Lösungsströme zur Zeit TN und RUA2Y die zur Zeit TN-DT sind (dies geschieht programmtechnisch innerhalb der Barrieren- schleife). Die relative Änderung der Aktivitätsströme wird nur berechnet, wenn die Nenner RUA1Y bzw. RUA2Y nicht verschwinden (verschwindet nur ein Nenner, entfällt die zweite Maximumbildung). Darüberhinaus wird das Maximum jeweils nur für solche Größen durchgeführt, für die gilt

$$|RU1Y| \cdot DT > ANTKM \cdot |RK1Y|$$

bzw.

$$|RU2Y| \cdot DT > ANTKM \cdot |RK2Y|$$

für alle Komponenten, d.h. der Beitrag des Stroms multipliziert mit DT größer ist als das ANTKM-fache der entsprechenden kumulierten Menge (ANTKM aus Job-Input-File: ZEITEN). Hierdurch wird z.B. vermieden, daß der Zerfall kurzlebiger Nuklide zu großen Zeiten die Zeitschrittweite beeinflusst, obwohl deren Aktivität auf unbedeutende Werte abgeklungen ist.

Zusätzlich erfolgt eine Meldung in der Ergebnisausgabe, welche Stromänderung in welcher Barriere den maximalen Wert EPS geliefert hat.

Berechnung einer neuen Zeitschrittweite (vgl. Teilaufgabe F im Ablaufdiagramm):

Das so ermittelte Maximum EPS wird mit einer vorgebenen Bandbreite für relative Änderungen verglichen,

$$\begin{aligned} & EPS1 < EPS < EPS2 \\ \text{wobei} \quad & EPS1 = EPSI \cdot (DDTU)^2 \\ \text{und} \quad & EPS2 = EPSI \cdot (DDTO)^2 \quad \text{gilt.} \end{aligned}$$

Die Größen EPSI, DDTU und DDTO werden über das Job-Input-File (ZEITEN) eingegeben.

Ist $EPS < EPS1$, so wird die Zeitschrittweite wie folgt vergrößert:

$$DT(\text{neu}) = DT(\text{alt}) \cdot DDTO \quad (DDTO > 1),$$

wenn sie nicht im Zeitschritt direkt davor schon vergrößert wurde.

Ist $EPS > EPS_2$, so wird die Zeitschrittweite wie folgt verkleinert:

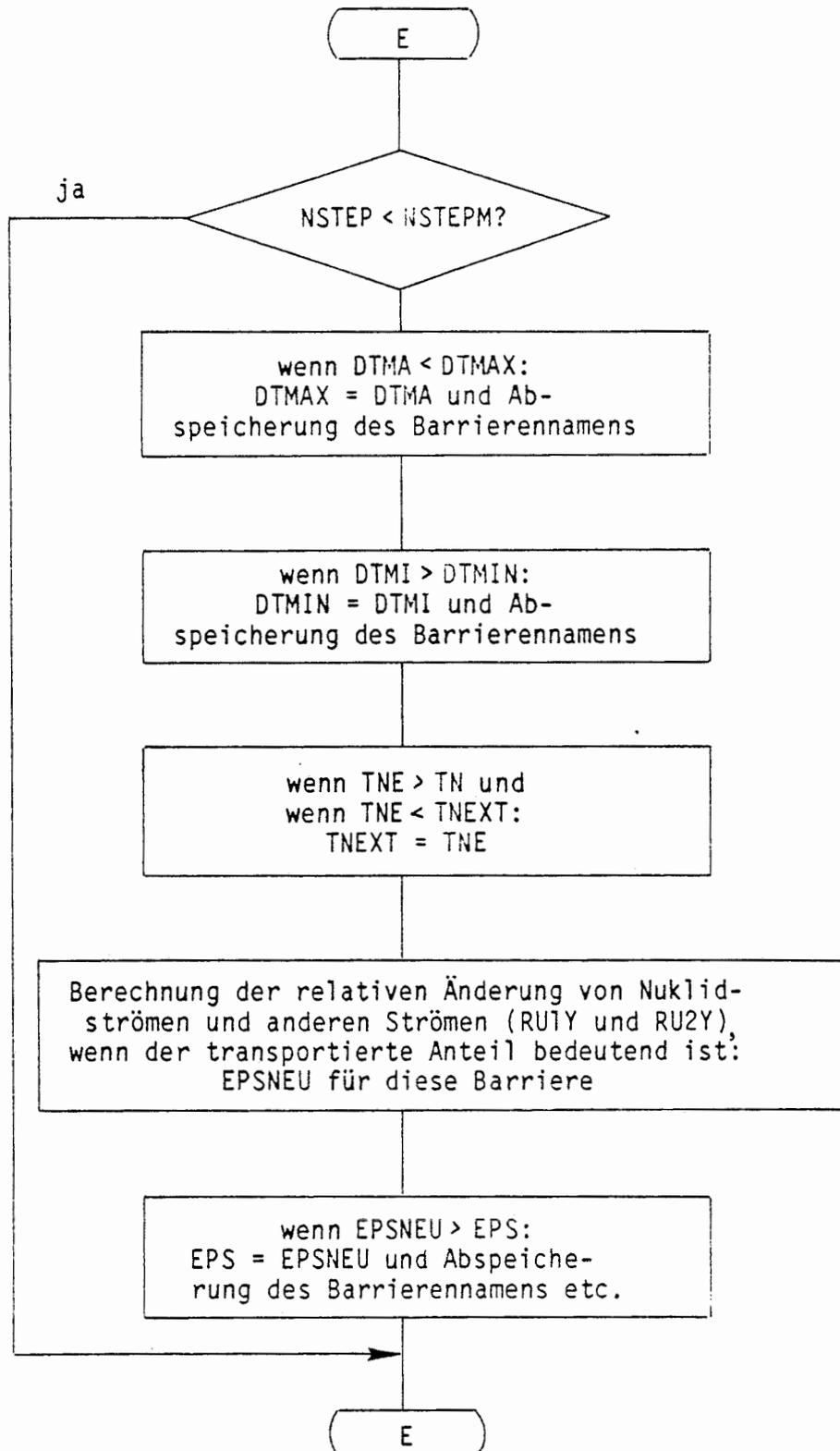
$$DT(\text{neu}) = DT(\text{alt}) \cdot DDTU \quad (DDTU < 1).$$

Die neue Zeitschrittweite wird ggf. nach oben so begrenzt, daß sie einen Wert DT_{MAX} nicht überschreitet, der sich aus den barrierenspezifischen Maximalwerten ergibt.

Die neue Zeitschrittweite wird ggf. nach unten so begrenzt, daß sie einen Wert DT_{MIN} nicht unterschreitet, der sich aus den barrierenspezifischen Minimalwerten (im allgemeinen DT_1) ergibt.

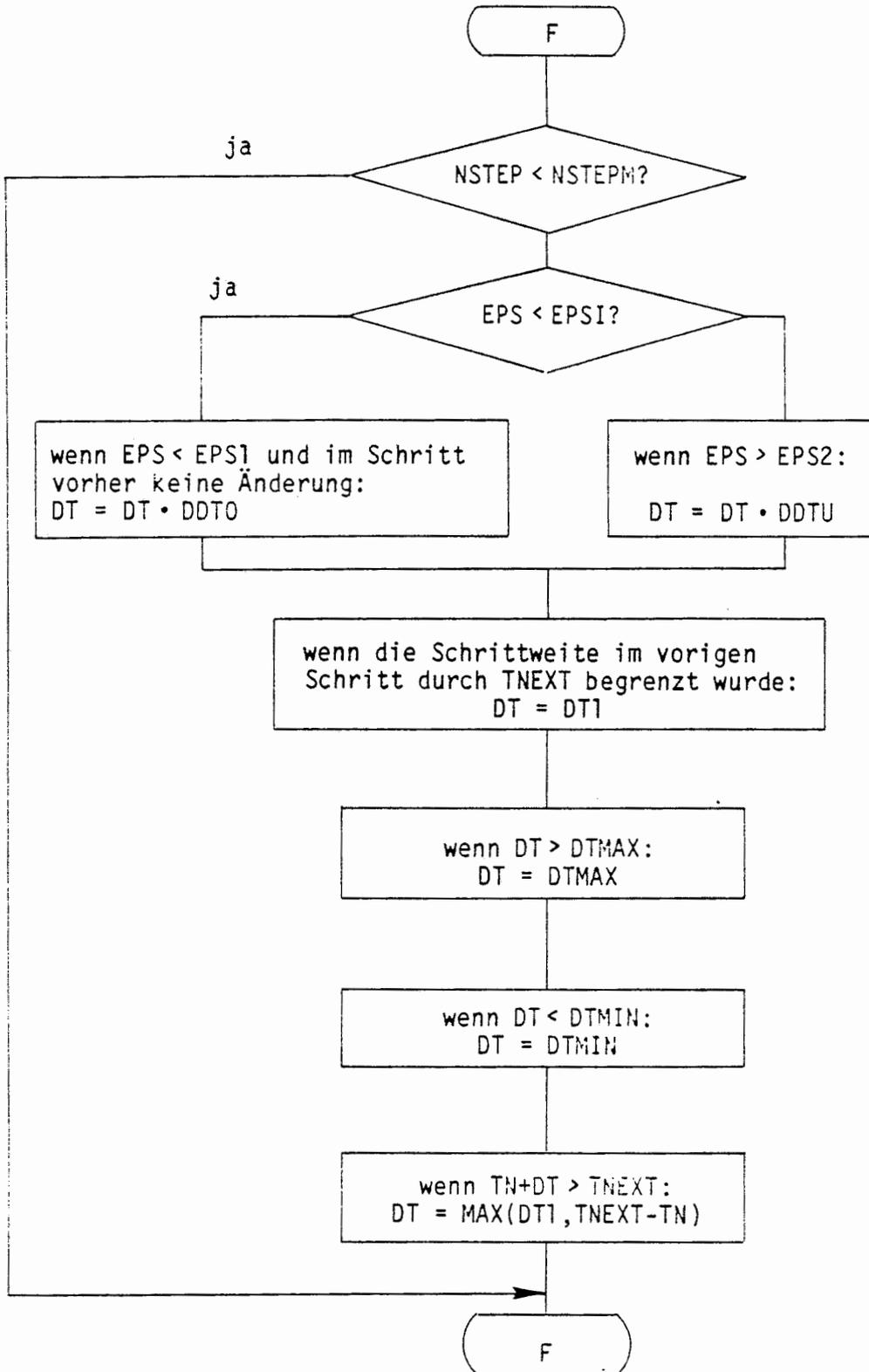
Weiterhin wird die Zeitschrittweite ggf. soweit reduziert (jedoch nicht unter DT_1), daß mit dem nächsten Schritt eine Zeit T_{NEXT} nicht überschritten wird, die das Minimum von aktuellen Vorgaben aus den Barrierenmodellen ist. Im Anschluß an eine solche Reduktion wird die nächstfolgende Schrittweite auf DT_1 festgelegt.

ABLAUF DES PROGRAMMS STEBA TEILAUFGABE E



In dieser Teilaufgabe ermittelte Größen:
Parameter für die Zeitschrittsteuerung (DTMAX, DTMIN, TNEXT, EPS)

ABLAUF DES PROGRAMMS STEBA TEILAUFGABE F



In dieser Teilaufgabe ermittelte Größen:
neue Zeitschrittweite (DT)

A 2.5.2 AUFRUF VON BARRIERENMODELLEN

Der Aufruf von Barrierenmodellen wird durch das Unterprogramm BMOD gesteuert. Dieses Programm sorgt für eine einheitliche Schnittstelle für alle Barrierenmodelle, d.h. die Liste der Übergangsparameter ist für alle Barrierenmodelle gleich, sowohl für solche geometrischer Barrieren, als auch für Mobilisierungsmodelle. Mit dem Unterprogrammaufruf erfolgt zugleich eine Umbenennung der Variablen in der Parameterliste, wobei die Komponenten von Matrizen (bzgl. Barrierenindex KBY) in Vektoren umgesetzt werden. Im einzelnen werden umgesetzt:

IB2Y(1, KBY)	in	IED1(1),	usw.
RB4Y(1,1, KBY)	in	DVN1(1,1),	usw.
RB3Y(1, KBY)	in	DV1 (1),	usw.
RB1Y(1, KBY)	in	ABD (1),	usw.
IB2Y(1, KBAY)	in	IED2(1),	usw.
RB4Y(1,1,KBAY)	in	DVN2(1,1),	usw.
RB3Y(1, KBAY)	in	DV2 (1),	usw.
RA1Y(1, KBY)	in	ZAN (1),	usw.
RA2Y(1, KBY)	in	ZAG (1),	usw.
RU1Y(1, KBY)	in	ZUN (1),	usw.
RU2Y(1, KBY)	in	ZUS (1),	usw.
RU3Y(1, KBY)	in	ZUD (1),	usw.

Aus programmtechnischen Gründen wird z.Z. das Feld DV1 der Parameterliste nicht weiter verarbeitet, stattdessen werden die geometrischen Barrierendaten in RB3Y im Programm BMOD aus dem COMMON-Block /WZBY/ in das Feld DVPI im COMMON-Block /E/ übertragen. Über diesen COMMON-Block werden sie den Barrierenmodellen zur Verfügung gestellt. Am Ende des Unterprogramms BMOD geschieht die Rückübertragung der (neu berechneten) Variablen in DVPI nach RB3Y. Der Aufruf BMOD im übergeordneten Programm STEBA erfolgt in jedem Zeitschritt für jede abgearbeitete Barriere.

Das Unterprogramm BMOD enthält eine DATA-Anweisung, die die Namen der verfügbaren Barrierenmodelle enthält. Bei Bedarf können einzelne Barrierenmodelle gegen neue ausgetauscht werden, oder es kann die Liste von Modellnamen entsprechend erweitert werden.

Der Aufruf eines bestimmten Barrierenmodells geschieht wie folgt:
Das Unterprogramm BMOD vergleicht den Namen eines Barrierenmodells, das im Job-Input-File einer Barriere zugewiesen wurde, mit den in der DATA-Anweisung vorhandenen Barrierenmodellnamen. Bei Übereinstimmung erfolgt ein Sprung zum Aufruf des entsprechenden Unterprogramms. Der universelle Aufruf lautet:

Call "Barrierenmodellname" (EVN, EVS, IED1, DVN1, ABD, IED2, DVN2, DV2, ZAN, ZAG, ZUN, ZUS, ZUD).

Nach Abarbeitung dieses Unterprogramms erfolgt Rücksprung an das übergeordnete Programm STEBA.

Falls keine Übereinstimmung der Modellnamen festgestellt wird, stoppt das Programm EMOS2 nach Aufruf der Hilfsroutine HRSTOP in BMOD.

ÜBERSICHT ÜBER DAS PROGRAMM BMOD

ÜBERGEORDNETES PROGRAMM :	STEBA
LÄNGE DES PROGRAMMS :	201 Zeilen 2687 Bytes
STAND DER DOKUMENTATION :	15. März 1986

KURZBESCHREIBUNG:

In einer DATA-Liste verfügbarer Barrierenmodelle wird das Modell für eine im Programm STEBA angesteuerte Barriere aufgesucht und das zugehörige Unterprogramm aufgerufen. Nach Abarbeitung des Barrierenmodells erfolgt Rücksprung nach STEBA (Programmabbruch durch HRSTOP, falls das erforderliche Barrierenmodell nicht in der Liste vorhanden ist).

PARAMETERLISTE:

CMOD, EVN(LNY), EVS(LRU2Y), IED1(LIB2Y), DVN1(LNY,LRB4Y), DV1(LRB3Y), ABD(LRB1Y), IED2(LIB2Y), DVN2(LNY,LRB4Y), DV2(LRB3Y), ZAN(LNY), ZAG(LRA2Y), ZUN(LNY), ZUS(LRU2Y), ZUD(LRU3Y)

EINGANGSPARAMETER:

CMOD - Name des Barrierenmodells
DV1 - Vektor geometrischer Barrierendaten (z.Z. nicht benutzt)

DURCHGANGSPARAMETER:

EVN - Eingangsvektor für Nuklidströme
EVS - Eingangsvektor für andere Ströme
IED1 - barrierenspezifische INTEGER-Eingangsdaten der aktuellen Barriere
DVN1 - Nukliddatenvektoren der aktuellen Barriere
ABD - Einlagerungszeit, Zuflußzeiten, Erreichen der Endporosität/Ende der Mobilisierung
IED2 - barrierenspezifische Integereingangsdaten der äußeren Barriere (z.Z. nicht benutzt)
DVN2 - Nukliddatenvektoren der äußeren Barriere
DV2 - geometrische Barrierendaten der äußeren Barriere
ZAN - Aktivitätsinventar der Barriere

ZAG - Matrix-, Behältermasse, Gebinde-, Hohlraum-
volumen, Zwischenlagerzeit
ZUN - Nuklidübergangsströme
ZUS - Übergabedaten Ströme
ZUD - sonstige Übergabedaten

LESENDER ZUGRIFF AUF COMMON-BLÖCKE:

/YZ/ : KB
/WZBY/ : RB2Y, RB3Y, IB2Y
/E/ : RED1, DVP1

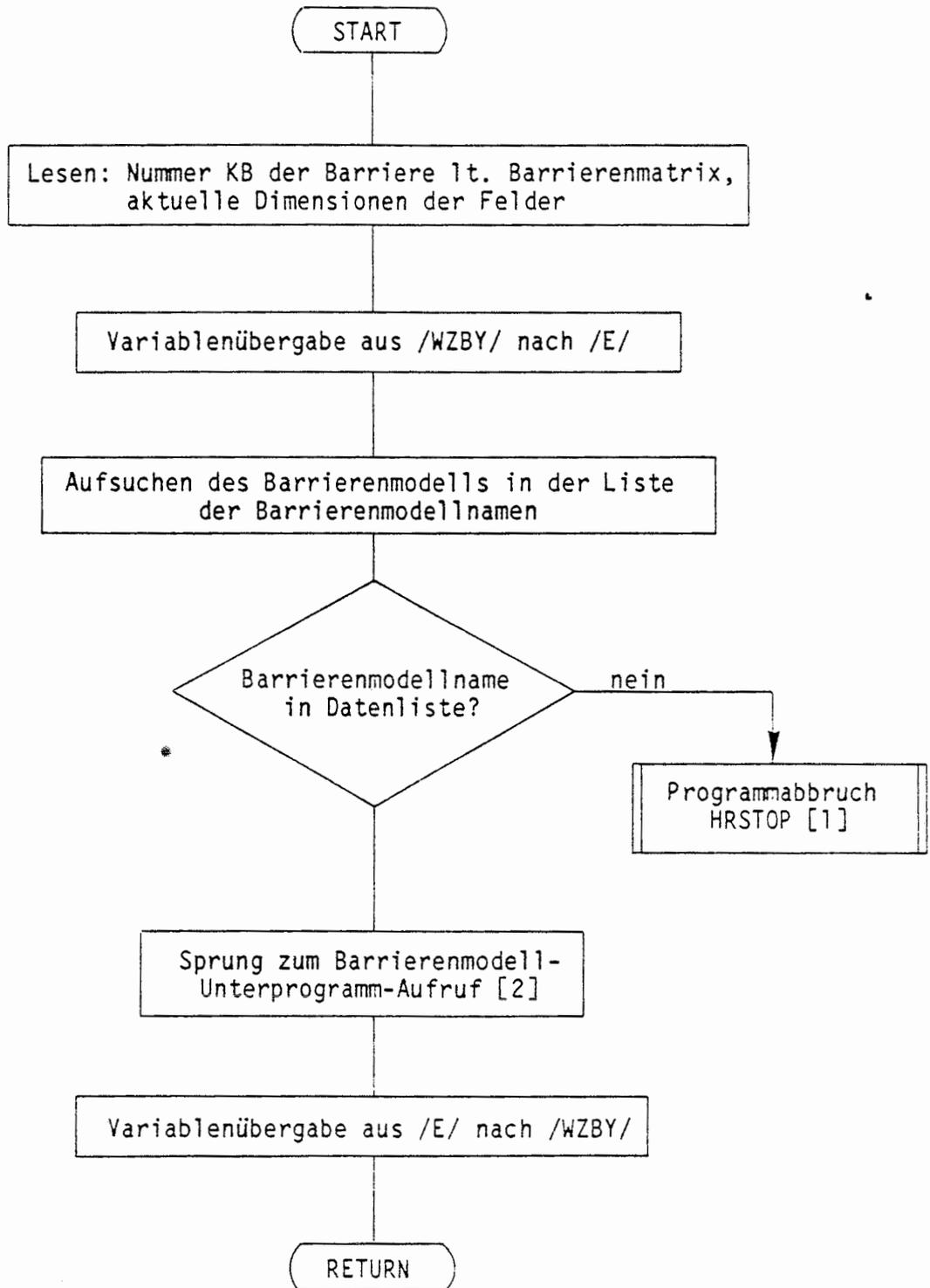
SCHREIBENDER ZUGRIFF AUF COMMON-BLÖCKE:

/WZBY/ : RB2Y, RB3Y, NRB2Y, NRB3Y
/WZUY : NRU2Y, NRU3Y
/E/ : RED1, DVP1

UNTERPROGRAMMAUFRUFE:

[1] HRSTOP('BMOD')
[2] Barrierenmodelle aus der Liste verfügbarer Modelle (bei
Bedarf zu ändern oder zu ergänzen)
Einheitliche Parameterliste: EVN, EVS, IED1, DVN1, ABD,
IED2, DVN2, DV2, ZAN, ZAG, ZUN, ZUS, ZUD)

ABLAUF DES PROGRAMMS BMOD



A 2.5.3 BARRIERENMODELLE

In einem Barrierenmodell wird der (nuklidspezifische) Aktivitätsausgangsstrom ZUN aus einer betrachteten Barriere berechnet. Er ergibt sich aus dem Aktivitätseingangsstrom EVN in die Barriere unter Berücksichtigung der physikalischen und chemischen Effekte (siehe A 2.5.4), die den Aktivitätsstrom durch die betrachtete Barriere wesentlich beeinflussen.

Die Modellierung der barriereninternen Vorgänge kann es erforderlich machen, daß neben den Aktivitätsströmen weitere Ein- bzw. Ausgangsströme betrachtet werden, beispielsweise Laugen- oder Gasströme. Derartige Ströme bilden die Komponenten des Eingangsvektors EVS bzw. des Ausgangsvektors ZUS.

Der Vektor ZAN enthält das aktuelle Aktivitätsinventar der betrachteten Barriere, das in jedem Zeitschritt unter Berücksichtigung der Ein- und Ausgangsströme sowie des Zerfalls bilanziert wird. Weitere nuklidspezifische Datensätze wie etwa die gelöste Aktivitätskonzentration können in dem Feld DVN1 abgelegt werden. Die Bedeutung der Spalten dieses Feldes kann für verschiedene Barrierenmodelle zum Studium barrierenspezifisch interessanter Nukliddatensätze unterschiedlich vereinbart werden.

Die Komponenten des Vektors ZUD dienen der Aufnahme von Größen, die im Rahmen der zeitabhängigen Ausgabe gedruckt werden sollen. Ihre Bedeutung kann ebenfalls barrierenspezifisch festgelegt werden.

Die restlichen Vektoren der genormten Parameterliste des Barrierenmodellaufrufes dienen der Datenversorgung des Unterprogramms. Die Bedeutung ihrer Komponenten ist größtenteils barrierenspezifisch und wird in den folgenden Unterkapiteln erläutert. Durch die barrierenspezifische Datenbeschickung des Modells wird erreicht, daß verschiedene Barrieren mit dem gleichen Barrierenmodell bearbeitet werden können.

Die Berechnung des Aktivitätstransportes in und aus einem Endlager mit Hilfe von Barrierenmodellen garantiert die große Flexibilität des Programmes EMOS2. Die Anpassung an neue Szenarien oder neue Endlagerformationen etc. erfolgt durch Änderung bzw. Austausch der entsprechenden Barrierenmodelle und Festlegung der neuen Barrierenstruktur im Job-Input-File.

A 2.5.3.1 ELEMENTSPEZIFISCHE MOBILISIERUNG (MOBZ6)

In dem Mobilisierungsmodell MOBZ6 wird der Aktivitätsstrom aus einem Abfallgebinde berechnet. Die Mobilisierung der Nuklide aus dem Gebinde wird durch den Behälter und die Fixierung in der Matrix behindert. In dem Modell wird angenommen, daß die Mobilisierung unmittelbar nach Ende der Betriebsphase beginnt und die elementspezifischen Mobilisierungsdauern mindestens ein Jahr betragen. Der Name MOBZ6 steht für die Version 6 eines Mobilisierungsmodells, das ursprünglich für die Mobilisierung aus zementierten Gebinden entwickelt wurde.

Innerhalb einer Schleife über alle Nuklide wird als erstes die relative Mobilisierungsrate r für jedes Nuklid berechnet. Ihr Zeitverlauf wird mit Hilfe der beiden Parameter maximale Behälterstandzeit t_B und elementspezifische Mobilisierungsdauer t_M beschrieben

für $t_M > t_B$,

$$r(t) = \begin{cases} 0 & \text{für } t \leq 0 \\ \frac{t}{t_B \cdot t_M} & \text{für } 0 < t < t_B \\ \frac{1}{t_M} & \text{für } t_B \leq t \leq t_M \\ \frac{t_B + t_M - t}{t_B \cdot t_M} & \text{für } t_M < t < t_M + t_B \\ 0 & \text{für } t_M + t_B \leq t \end{cases}$$

und für $t_M \leq t_B$

$$r(t) = \begin{cases} 0 & \text{für } t \leq 0 \\ \frac{t}{t_B \cdot t_M} & \text{für } 0 < t < t_M \\ \frac{1}{t_B} & \text{für } t_M \leq t \leq t_B \\ \frac{t_B + t_M - t}{t_B \cdot t_M} & \text{für } t_B < t < t_B + t_M \\ 0 & \text{für } t_B + t_M \leq t \end{cases}$$

Der Parameter t_B steht dem Unterprogramm als RED1(2) über den COMMON-Block /E/ zur Verfügung, während die elementspezifischen Mobilisierungsdauern als eine Zeile der Matrix REY im COMMON-Block /WZEY/ übergeben werden. Die Nummer der Zeile kann als IED1(16) der Parameterliste entnommen werden. Diese Zeilennummer ist barrierenspezifisch zur Berücksichtigung verschiedener Fixierungsarten, die in Mobilisierungsgruppen zusammengefaßt sind.

Bis zum Ablauf der Zeit $t_B + t_M$ ergibt sich der Aktivitätsausgangstrom als Produkt der relativen Mobilisierungsrate mit einem hypothetischen Aktivitätsinventar in dem Gebinde, das als nuklidspezifischer Datensatz in DVN1 abgelegt ist. Zu Beginn der zeitabhängigen Rechnung stimmt dieses hypothetische Aktivitätsinventar mit dem wahren Inventar des betrachteten Gebindes überein. In der Folge wird es lediglich durch den Zerfall, nicht aber durch die Mobilisierung verändert.

Nach Ablauf der Zeit $t_B + t_M$ ist die Mobilisierung für Spalt- und Aktivierungsprodukte abgeschlossen, weil zu diesem Zeitpunkt alle Nuklide der betreffenden Elemente mobilisiert sind. Radionuklide aus Zerfallsreihen können auch nach Ablauf ihrer Mobilisierungsdauer noch in den Gebinden vorhanden sein, wenn sie aus anderen Radionukliden mit größerer Mobilisierungsdauer nachgebildet werden. Der Aktivitätsausgangstrom wird in diesem Zeitbereich als $1/t_M$ -ter Teil des vorhandenen Aktivitätsinventars berechnet.

ÜBERSICHT ÜBER DAS PROGRAMM MOBZ6

ÜBERGEORDNETES PROGRAMM : BMOD

LÄNGE DES PROGRAMMS : 323 Zeilen
 1126 Bytes

STAND DER DOKUMENTATION : 15. März 1986

KURZBESCHREIBUNG:

Berechnung des Aktivitätsausgangstromes aus einem Abfallgebinde unter Berücksichtigung elementspezifischer Mobilisierungsdauern.

PARAMETERLISTE:

EVN(LNY), EVS(LRU2Y),
IED1(LIB2Y), DVN1(LNY,LRB4Y), ABD(LRB1Y)
IED2(LIB2Y), DVN2(LNY,LRB4Y), DV2(LRB3Y),
ZAN(LNY), ZAG(LRA2Y), ZUN(LNY), ZUS(LRU2Y), ZUD(LRU3Y)

EINGANGSPARAMETER:

- IED1(1) : Anzahl der Komponenten des Vektors RED1 im COMMON-Block
/E/
IED1(2) : Anzahl nuklidspezifischer Datensätze in DVN1
IED1(3) : Anzahl der Komponenten des Vektors DVPI im COMMON-Block
/E/
IED1(16) : Nummer der Zeile in der Matrix REY im COMMON-Block
/WZEY/, die die elementspezifischen Mobilisierungsdauern enthält
IED1(17) : Barrierenspezifischer Schalter
=1: In einer Zerfallsreihe des Inventars dieses Gebindes
gibt es Nuklide, deren zugehörige Elemente unterschiedliche
Mobilisierungsdauern haben
=0: sonst
ABD(1) : Einlagerungszeitpunkt des Gebindes, bezogen auf den Beginn
der Betriebsphase [a]
ABD(4) : Ende der Mobilisierung aus dem Gebinde, bezogen auf den
Beginn der Nachbetriebsphase [a]
DVN1(KNY,1) : Hypothetisches Aktivitätsinventar des Gebindes zu Beginn
des Zeitschrittes [Bq]
ZAN : Aktivitätsinventar des Gebindes zu Beginn des Zeitschrittes
[Bq]

AUSGANGSPARAMETER:

- ABD(4) : Ende der Mobilisierung aus dem Gebinde, bezogen auf den
Beginn der Nachbetriebsphase [a]
DVN1(KNY,1) : Hypothetisches Aktivitätsinventar des Gebindes am Ende
des Zeitschrittes [Bq]
ZAN : Aktivitätsinventar des Gebindes am Ende des Zeitschrittes
[Bq]
ZUN : Aktivitätsausgangsstrom [Bq/a]
ZUD : Andere Übergabedaten für die Ausgabe zu Kontrollzwecken

LESENDER ZUGRIFF AUF COMMON-BLÖCKE:

/E/ : RED1
/YZ/ : NSTEP, TN, TB
/RZ/ : IDR(5)
/WZNY/ : INY(4,KNY), NNY, NNRY(1)
/WZEY/ : REY(IED1(16),.), NEY
/G/ : G(54)
/WZUY/ : NRU3Y

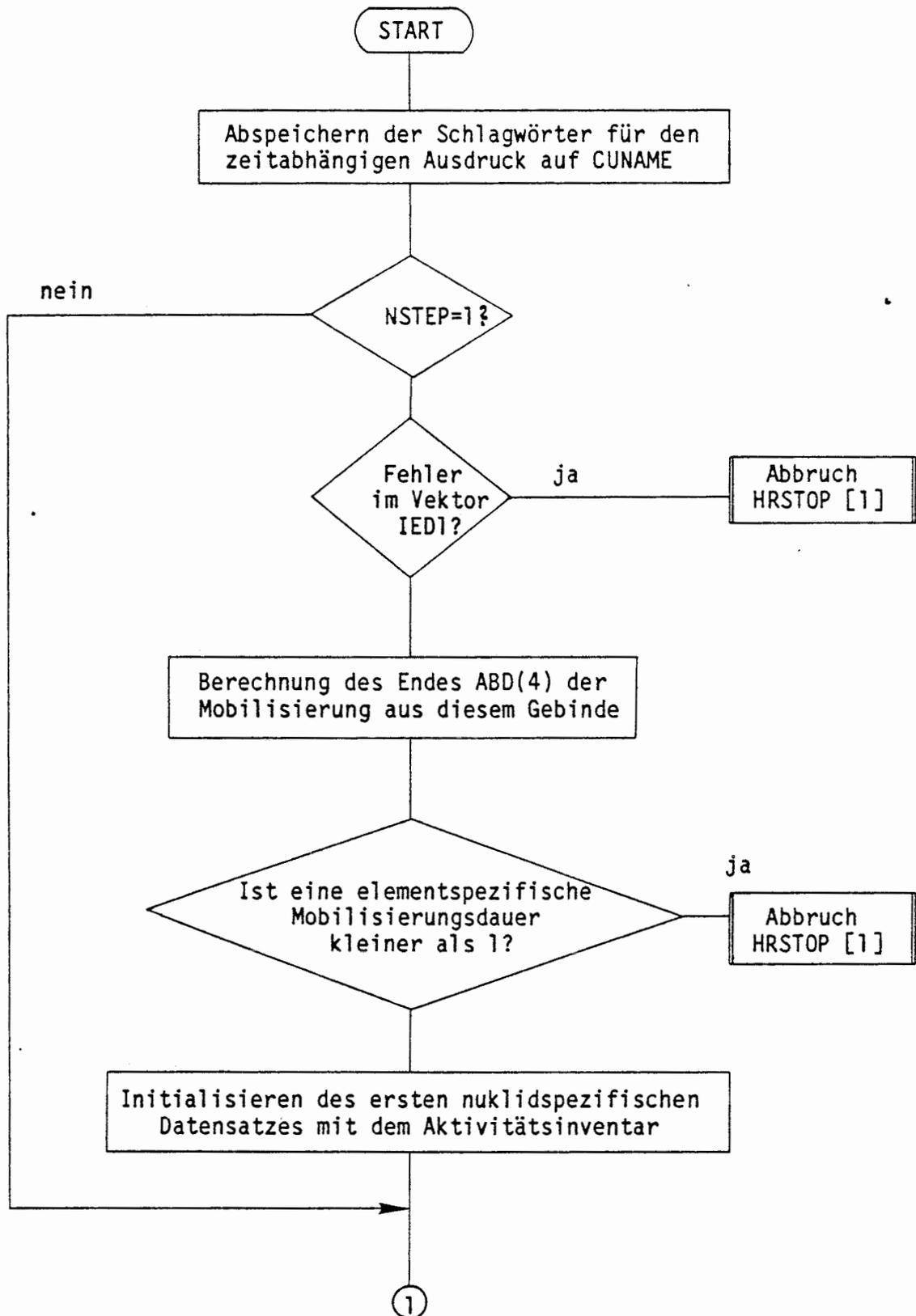
SCHREIBENDER ZUGRIFF AUF COMMON-BLÖCKE:

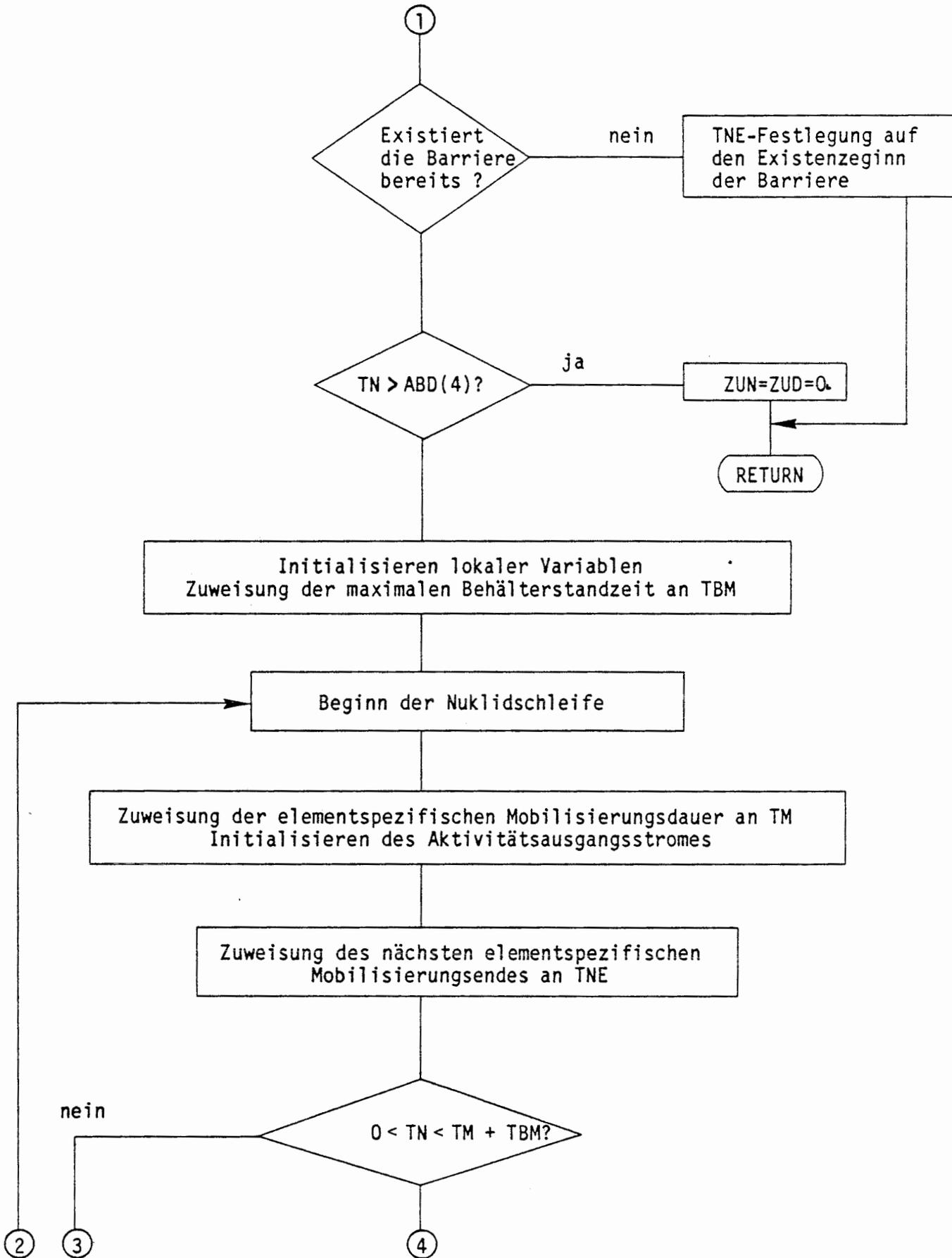
/YZ/ : TNE
/N/ : CUNAME

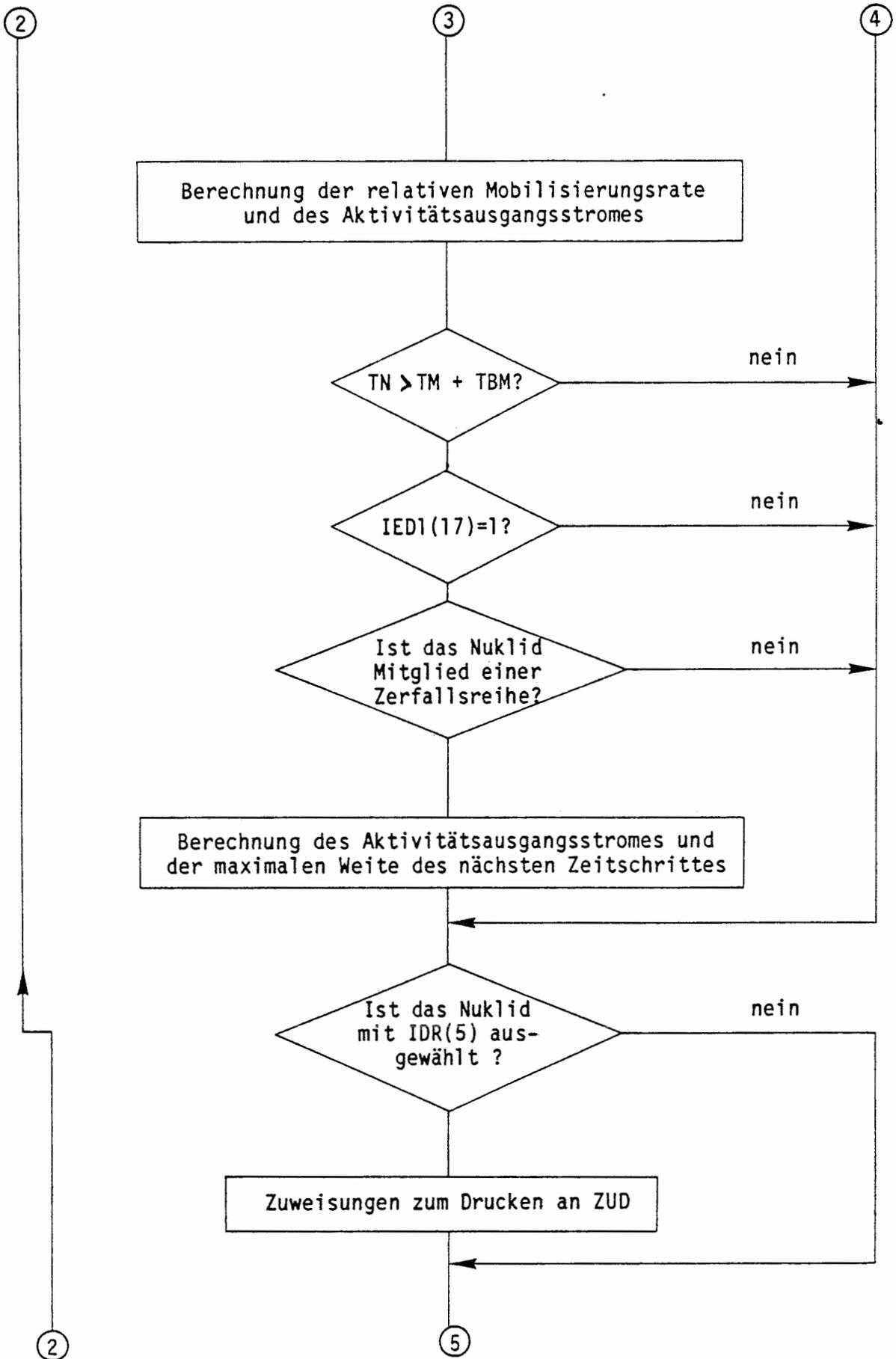
UNTERPROGRAMMAUFRUFE:

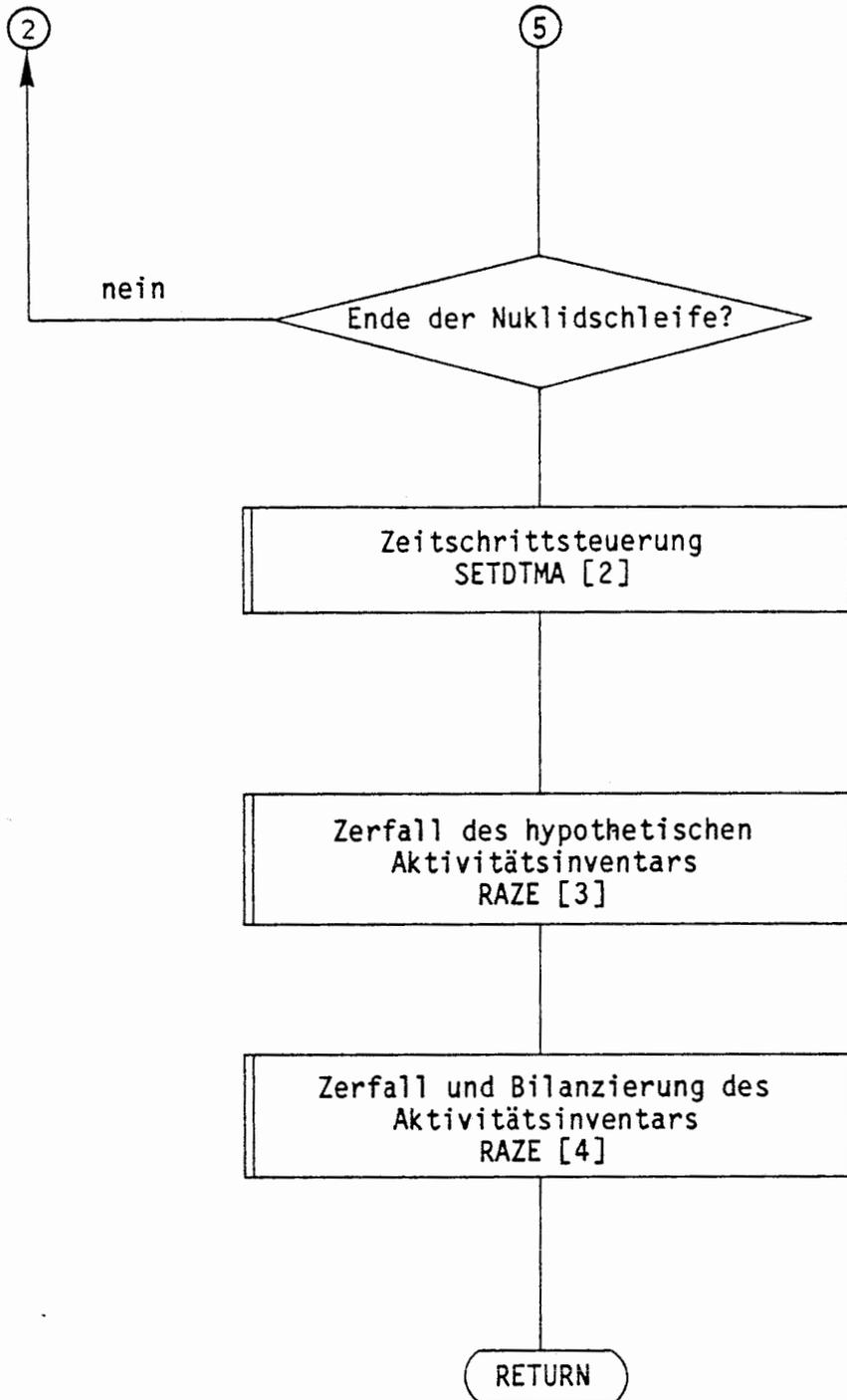
[1] HRSTOP ('MOBZ6')
[2] SETDTMA (DTMAN, 'MOBZ6')
[3] RAZE (0, DVN1, ZERO, ZERO)
[4] RAZE (1, ZAN, ZERO, ZERO)

ABLAUF DES PROGRAMMS MOBZ6









A 2.5.3.2 VOLUMENQUELLE (VQUELLE1)

In dem Barrierenmodell VQUELLE1 wird der Aktivitätsstrom aus einem Modellvolumen berechnet, das von einem konstanten Lösungsvolumenstrom durchflossen wird. Der Aktivitätsausgangsstrom ergibt sich als Produkt dieses Lösungsvolumenstromes mit der Aktivitätskonzentration, die zu Beginn des Zeitschrittes in dem Modellvolumen in gelöster Form vorhanden ist.

Mit Hilfe des Unterprogrammes RAZE wird die Änderung des Aktivitätsinventars in dem Modellvolumen in dem aktuellen Zeitschritt berechnet. Gleichzeitig wird das Aktivitätsinventar unter Berücksichtigung des Aktivitätseingangs- und Ausgangsstromes bilanziert.

Die gelöste Aktivität in dem Modellvolumen am Ende des Zeitschrittes wird anschließend in dem Unterprogramm SORPRP2 unter Berücksichtigung der Ausfällung und der Sorption berechnet. Bezieht man die gelöste Aktivität auf das vorhandene Lösungsvolumen, so erhält man die gelöste Aktivitätskonzentration. Aus dieser ergibt sich durch Multiplikation mit dem Dosiskonversionsfaktor die Dosis, der die Biosphäre bei Nutzung dieser Lösung ausgesetzt wäre.

ÜBERSICHT ÜBER DAS PROGRAMM VQUELLE1

ÜBERGEORDNETES PROGRAMM :	BMOD
LÄNGE DES PROGRAMMS :	182 Zeilen 549 Bytes
STAND DER DOKUMENTATION :	15. März 1986

KURZBESCHREIBUNG:

Berechnung des Aktivitätsausgangsstromes aus einem Modellvolumen, das von einem konstanten Lösungsvolumenstrom durchflossen wird, unter Berücksichtigung von Zerfall, Sorption und Ausfällung.

PARAMETERLISTE:

EVN(LNY), EVS(LRU2Y),
IED1(LIB2Y), DVN1(LNY, LRB4Y), ABD(LRB1Y),
IED2(LIB2Y), DVN2(LNY, LRB4Y), DV2(LRB3Y),
ZAN(LNY), ZAG(LRA2Y), ZUN(LNY), ZUS(LRU2Y), ZUD(LRU3Y)

EINGANGSPARAMETER:

EVN : Aktivitätseingangsstrom [Bq/a]
IED1(1) : Anzahl der Komponenten des Vektors RED1 im COMMON-Block /E/
IED1(2) : Anzahl der nuklidspezifischen Datensätze in DVN1

IED1(3) : Anzahl der Komponenten des Vektors DVPI im COMMON-Block /E/
IED1(14) : Nummer der Zeile in der Matrix REY im COMMON-Block /WZEY/, die die elementspezifischen Löslichkeitsgrenzen enthält
IED1(15) : Nummer der Zeile in der Matrix REY im COMMON-Block /WZEY/; die die elementspezifischen KD-Werte enthält
DVN1(KNY,1) : Gelöste Aktivität im Modellvolumen zu Beginn des Zeitschrittes [Bq]
ABD(1) : Zeitpunkt des Abwerfens der Grubenteile, die das Modellvolumen bilden, bezogen auf den Beginn der Betriebsphase [a]
ZAN : Aktivitätsinventar im Modellvolumen zu Beginn des Zeitschrittes [Bq]

AUSGANGSPARAMETER:

DVN1(KNY,1) : Gelöste Aktivität im Modellvolumen am Ende des Zeitschrittes [Bq]
DVN1(KNY,2) : Aktivitätskonzentration im Modellvolumen am Ende des Zeitschrittes [Bq/m³]
DVN1(KNY,3) : Aus DVN1(KNY,2) resultierende Dosis [Sv/a]
ZAN : Aktivitätsinventar im Modellvolumen am Ende des Zeitschrittes [Bq]
ZUN : Aktivitätsausgangsstrom aus dem Modellvolumen [Bq/a]
ZUD(1) : Gelöste Konzentration des IDR(5)-ten Nuklides im Modellvolumen [mol/l]
ZUD(2) : Lösungsvolumen, das im aktuellen Zeitschritt das Modellvolumen durchfließt [m³]
ZUD(3) : Kumuliertes Lösungsvolumen, das das Modellvolumen durchflossen hat [m³]
ZUD(4) : Aktivitätseingangsstrom des IDR(5)-ten Nuklides [Bq/a]

LESENDER ZUGRIFF AUF COMMON-BLÖCKE:

/E/
/G/ : G(48)
/WZNY/ : RNY(3,KNY),
/YZ/ : DT, NSTEP, TB, TN

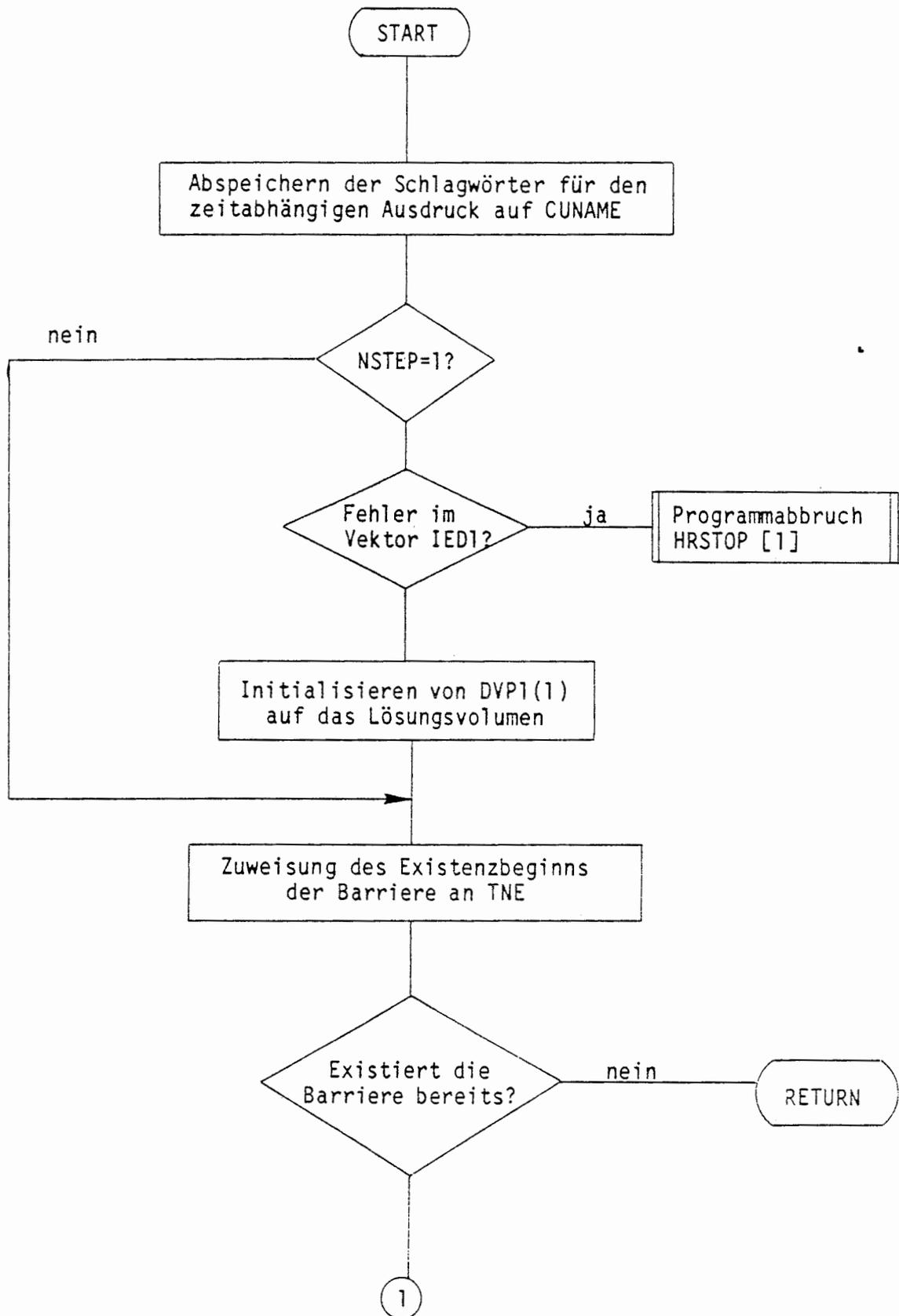
SCHREIBENDER ZUGRIFF AUF COMMON-BLÖCKE:

/YZ/ : TNE
/N/ : CUNAME
/E/ : DVPI(1)=VLZ

UNTERPROGRAMMAUFRUFE:

[1] HRSTOP ('VQUELLE1')
[2] RAZE (1, ZAN, EVN, ZUN)
[3] SORPRP2(1, VL, MSORBENS, IED1(14), IED1(15), ZAN, DVN1(1,1), ZUD)

ABLAUF DES PROGRAMMS VQUELLE1





A 2.5.4 EINZELEFFEKTE

Die Größe des Aktivitätsstromes aus den verschiedenen Barrieren wird durch eine Reihe von physikalischen oder chemischen Einzeleffekten beeinflusst. Die Auswirkungen dieser Einzeleffekte werden in eigenständigen Unterprogrammen behandelt. Dies erleichtert die Anpassung dieser Programmteile an den aktuellen Stand von Wissenschaft und Technik und die Berücksichtigung bisher nicht betrachteter Einzeleffekte. Der Aufruf der Unterprogramme, die Einzeleffekte behandeln, erfolgt in Barrierenmodellen oder anderen Einzeleffekten.

A 2.5.4.1 RADIOAKTIVER ZERFALL UND BILANZIERUNG (RAZE)

In dem Unterprogramm RAZE (RADioaktiver ZERfall) wird das Aktivitätsinventar DV am Ende eines Zeitschrittes unter Berücksichtigung des radioaktiven Zerfalls und (bei Schalter ISW=1) des Aktivitätseingangs- und -ausgangsstromes berechnet. Die Differentialgleichung, die die Bilanz beschreibt, wird zur zeitdiskreten Behandlung in eine Differenzgleichung umgeschrieben.

$$DV(t+DT) = \frac{DV(t) \cdot \left(1 - \frac{\lambda \cdot DT}{2}\right) + DIF + DT \cdot (SNI - SNO)}{1 + \frac{\lambda \cdot DT}{2}}$$

Die Zerfallskonstanten werden als Vektor RNY(2, KNY) über den COMMON-Block /WZNY/, die Zeitschrittweite DT über den COMMON-Block /YZ/ übergeben. Der Summand DIF beschreibt den Aktivitätszuwachs, den das betrachtete Nuklid durch den Zerfall der Mütter erfährt. Bei n Müttern gilt:

$$DIF = DT \sum_{i=1}^n \frac{DV_i(t) + DV_i(t+DT)}{2}$$

AUSGANGSPARAMETER:

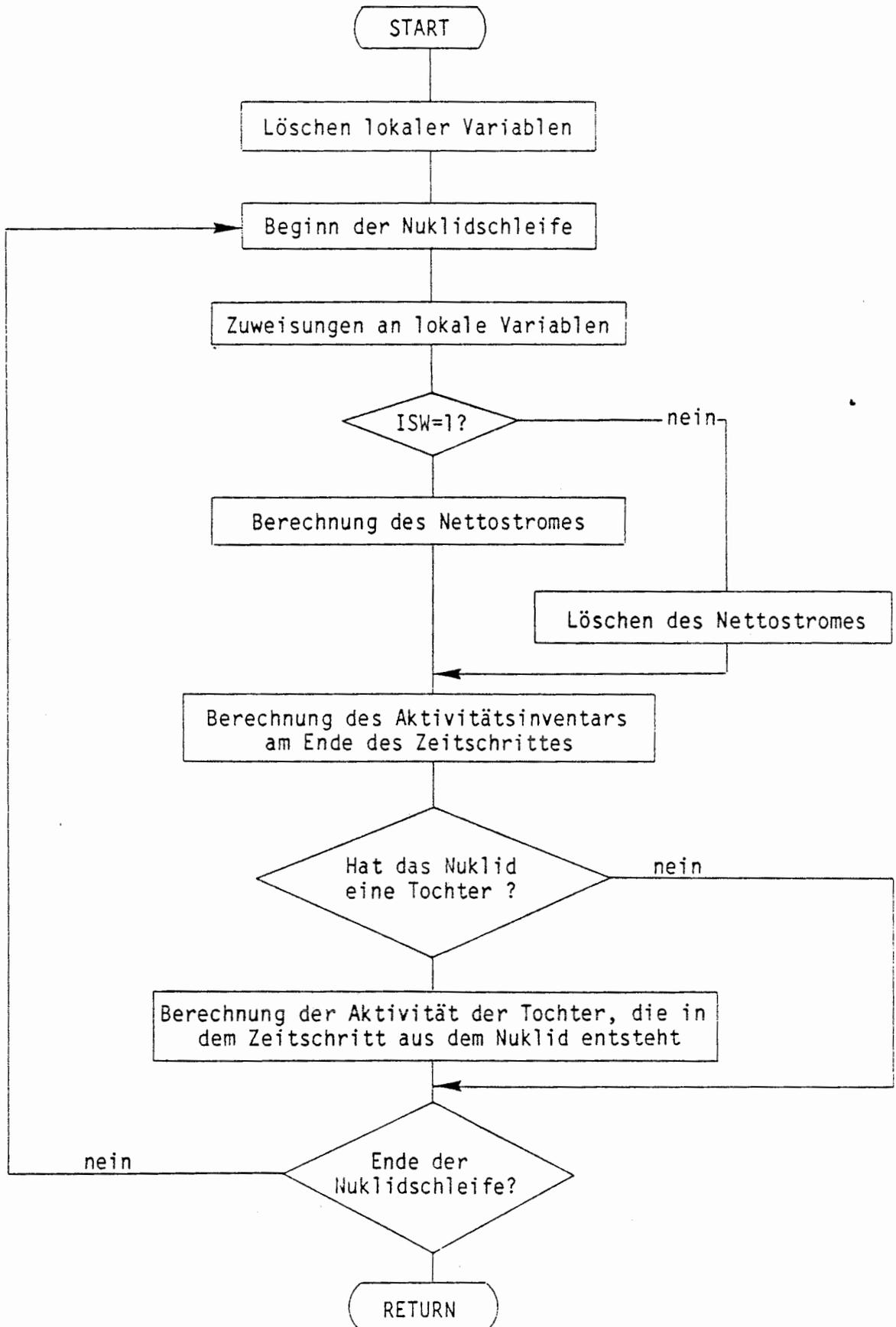
DV : Aktivitätsinventar am Ende des Zeitschrittes [Bq]

LESENDER ZUGRIFF AUF COMMON-BLÖCKE:

/YZ/ : DT

/WZNY/

ABLAUF DES PROGRAMMS RAZE



A 2.5.4.2 SORPTION UND AUSFÄLLUNG (SORPRP2)

In dem Unterprogramm SORPRP2 (SORption und Ausfällung, engl. PREciPitation) wird die aktuelle gelöste Aktivität in einer Barriere berechnet. Sie ergibt sich als ein Teil der insgesamt in der Barriere vorhandenen Aktivität, dessen Größe durch die elementspezifische Sorption und Ausfällung bestimmt wird. Zusätzlich (ISW=1) besteht die Möglichkeit, die Zeitpunkte in Tabellen zu schreiben, zu denen maximale Elementkonzentrationen und erstmalig Löslichkeitsgrenzen erreicht werden.

Mit der Option NKL=0 hat man die Möglichkeit, ohne Sorption zu rechnen. Liegt dann die Konzentration des Elementes in der vorhandenen Lösung unterhalb der Löslichkeitsgrenze, ist der gelöste Anteil identisch 1. Andernfalls entspricht er dem Quotienten aus maximal löslicher und insgesamt vorhandener Stoffmenge des Elementes.

Unter Berücksichtigung der Sorption entspricht der gelöste Anteil ebenfalls vorstehendem Quotienten, falls Ausfällung stattfindet. Andernfalls ergibt er sich aus dem KD-Konzept zu

$$\frac{1}{1 + \frac{m_s \cdot KD}{V_L \cdot 1000}} ,$$

wobei die sorbierende Masse m_s g und das Lösungsvolumen V_L über die Parameterliste zur Verfügung stehen, während die KD-Werte als NKL-te Zeile der Matrix REY im COMMON-Block /WZEY/ übergeben werden.

ÜBERSICHT ÜBER DAS PROGRAMM SORPRP2

ÜBERGEORDNETES PROGRAMM : BMOD

LÄNGE DES PROGRAMMS : 179 Zeilen
579 Bytes

STAND DER DOKUMENTATION : 15. März 1986

KURZBESCHREIBUNG:

Aus dem aktuellen Aktivitätsinventar in einer Barriere wird unter Berücksichtigung der elementspezifischen Sorption und Ausfällung das aktuelle gelöste Aktivitätsinventar berechnet. Auf Wunsch werden die Zeitpunkte in Tabellen geschrieben, zu denen maximale Elementkonzentrationen und erstmalig Löslichkeitsgrenzen erreicht werden.

PARAMETERLISTE:

ISW, VL, MSORBENS, NLG, NKL, IAS(LNY), IAL(LNY), ZUD(LRU3Y)

EINGANGSPARAMETER:

ISW : Schalter
=0 : Berechnung der gelösten Aktivität
=1 : Wie 0 und zusätzlich Eintrag in die Tabellen

VL : Lösungsvolumen [m³]

MSORBENS : Sorbierende Masse [kg]

NLG : Nummer der Zeile in der Matrix REY im COMMON-Block /WZEY/, in der die elementspezifischen Löslichkeitsgrenzen stehen

NKL : Nummer der Zeile in der Matrix REY im COMMON-Block /WZEY/, in der die elementspezifischen KD-Werte stehen

IAS : Gesamtaktivitätsinventar in der betrachteten Barriere [Bq]

AUSGANGSPARAMETER:

IAL : gelöstes Aktivitätsinventar [Bq]
ZUD(1) : Konzentration des IDR(5)-ten Nuklides in Lösung [mol/l]

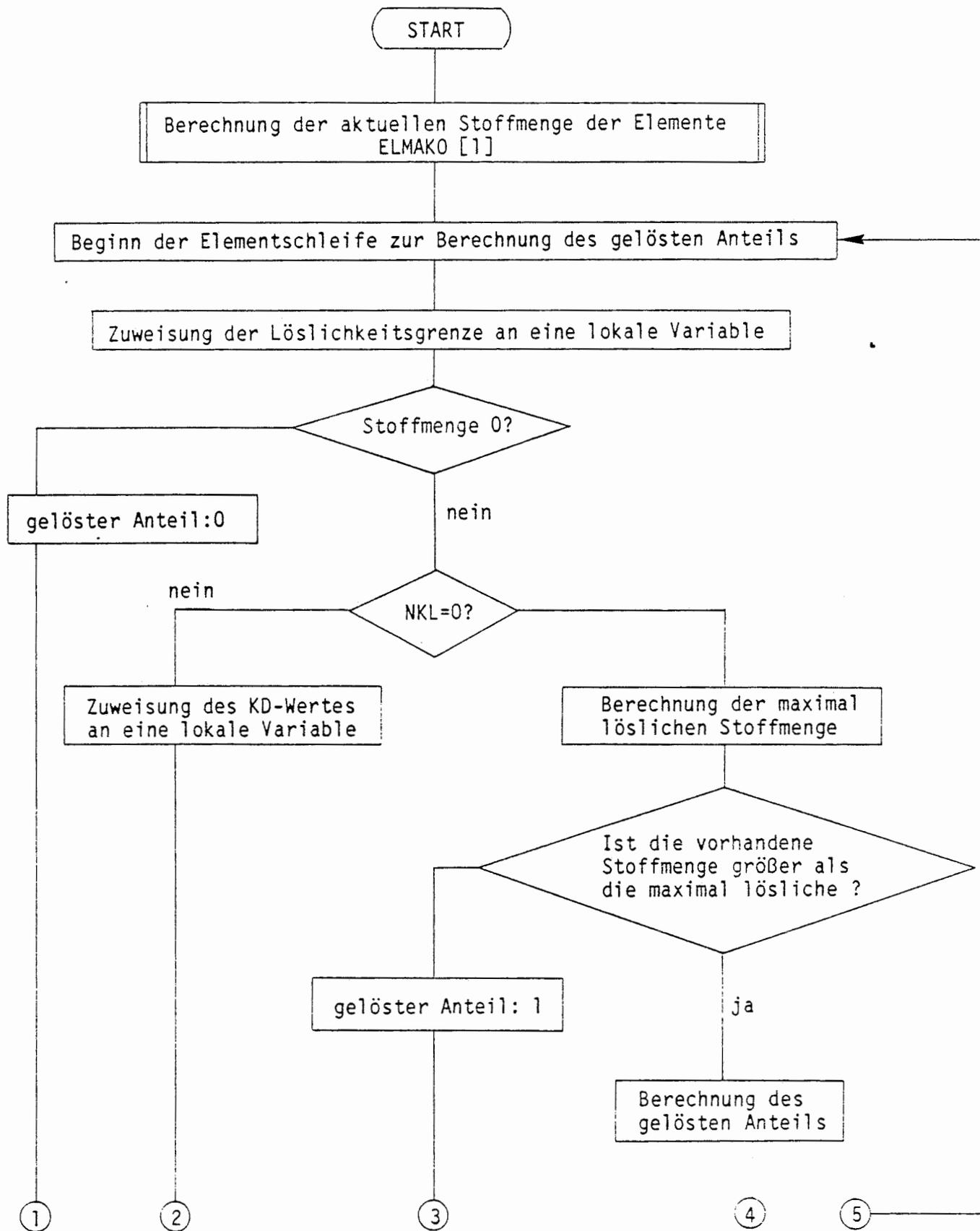
LESENDER ZUGRIFF AUF COMMON-BLÖCKE

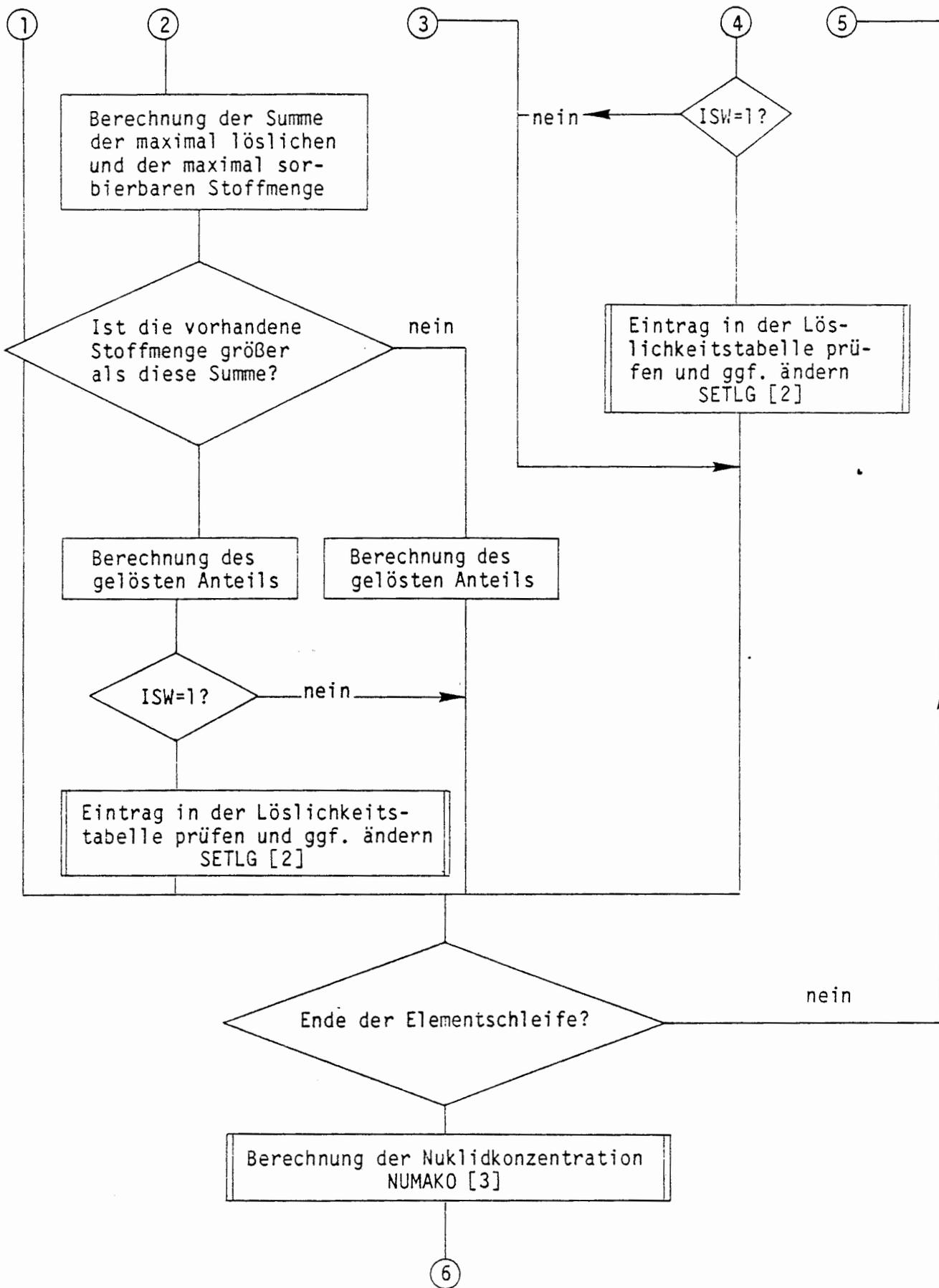
/WZEY/ : NEY, REY
/G/ : G(51), G(52)
/WZNY/ : INY(4,KNY)
/RZ/ : IDR(5)

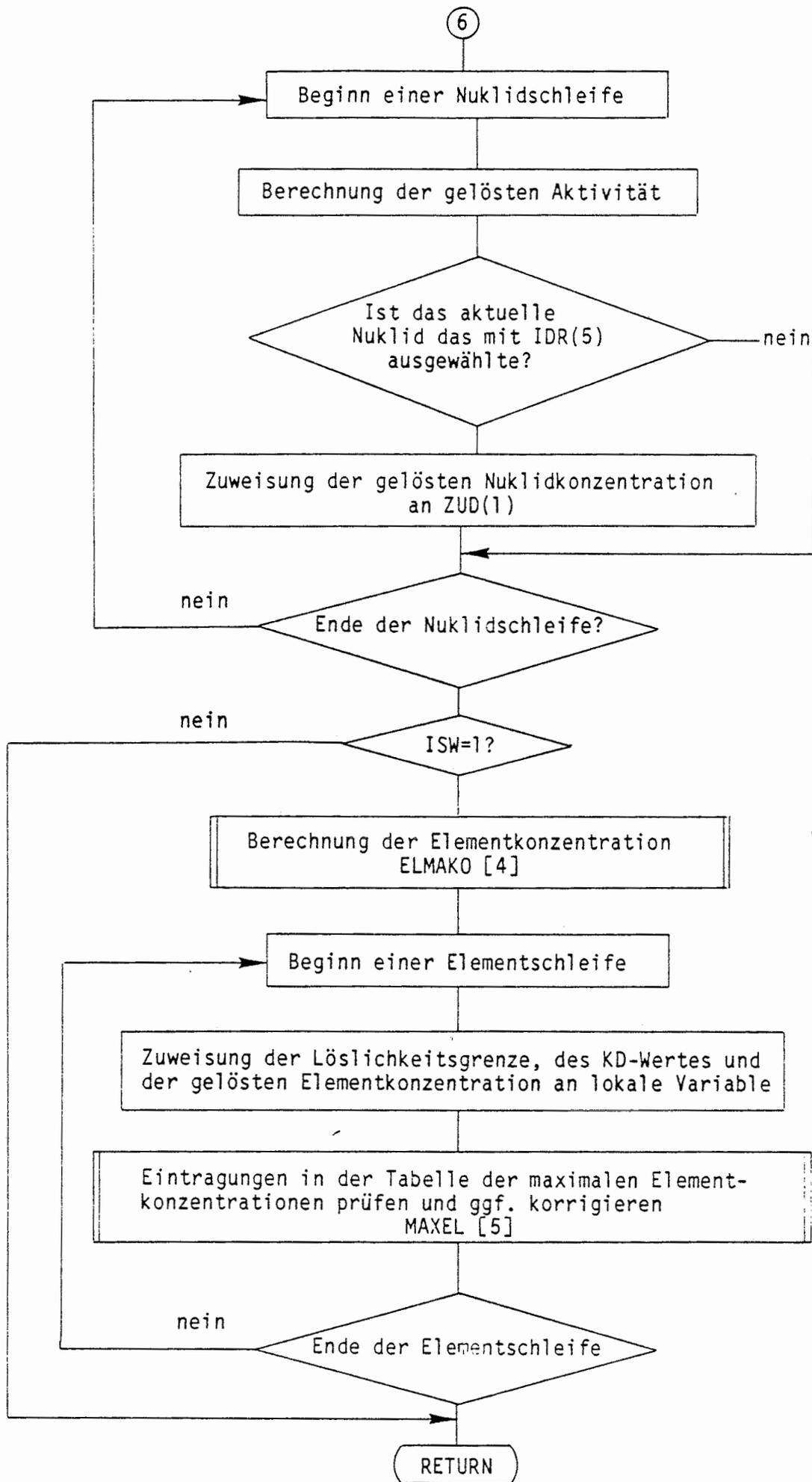
UNTERPROGRAMMAUFRUFE:

[1] ELMAKO (3, VL, IAS(1), IELMSUM(1))
[2] SETLG(KEY)
[3] NUMAKO (2, VL, IAS(1), CNUKLID(1))
[4] ELMAKO (2, VL, IAS(1), CELEMEN(1))
[5] MAXEL (KEY, CELEMEN(KEY), LG, QU, KL, ANTEIL(KEY), CL)

ABLAUF DES PROGRAMMS SORPRP2







A 2.5.5 HILFSROUTINEN

In diesem Unterkapitel werden Hilfsroutinen beschrieben, die in Barrierenmodellen oder Einzeleffekten aufgerufen werden.

A 2.5.5.1 UMRECHNUNG VON AKTIVITÄTEN (ELMAKO)

Die Hilfsroutine ELMAKO (ELEMENT-Massen bzw. -Konzentrationen) dient dazu, ein gegebenes nuklidspezifisches Aktivitätsinventar in Massen, Konzentrationen oder Stoffmengen umzurechnen. Während diese Größen bei dem ENTRY ELMAKO für die vorhandenen Elemente ermittelt werden, kann man sie bei Benutzung des Entries NUMAKO (NUKlid-Massen bzw. Konzentrationen) nuklidspezifisch berechnen. Art und Einheit der Ausgangsgröße werden mit dem Schalter ISW gesteuert.

Die Umrechnung erfolgt durch Multiplikation der Aktivitäten mit den Massenkonzentrationsfaktoren, die als 5. Zeile der Matrix RNY im COMMON-Block /WZNY/ verfügbar sind. Für ISW=1 und ISW=2 erfolgt anschließend eine Division durch das Lösungsvolumen VL. Für ISW=2 bzw. ISW=3 wird eine weitere Division durch das Atomgewicht in g/mol bzw. kg/mol durchgeführt, das als 1. Zeile in der Matrix INY im COMMON-Block /WZNY/ übergeben wird. Bei Benutzung der Entries ELMAKO werden die Ergebnisse für die Nuklide eines Elementes addiert.

ÜBERSICHT ÜBER DAS PROGRAMM ELMAKO/NUMAKO

ÜBERGEORDNETES PROGRAMM :	BMOD
LÄNGE DES PROGRAMMS :	126 Zeilen 398 Bytes
STAND DER DOKUMENTATION :	15. März 1986

KURZBESCHREIBUNG:

Umrechnung von Aktivitäten in Massen, Konzentrationen oder Stoffmengen.

PARAMETERLISTE:

ISW, VL, DVI(LNY), DVO(LNY)

EINGANGSPARAMETER:

ISW : Schalter

=0 : Berechnung von Element-/Nuklid-Massen [kg]
=1 : Berechnung von Element-/Nuklid-Konzentrationen [kg/m³]
=2 : Berechnung von Element-/Nuklid-Konzentrationen [mol/l]
=3 : Berechnung von Element-/Nuklid-Stoffmengen [mol]

VL : Lösungsvolumen [m³]
DVI : nuklidspezifisches Aktivitätsinventar [Bq]

AUSGANGSPARAMETER:

DVO : entsprechend dem Schalter

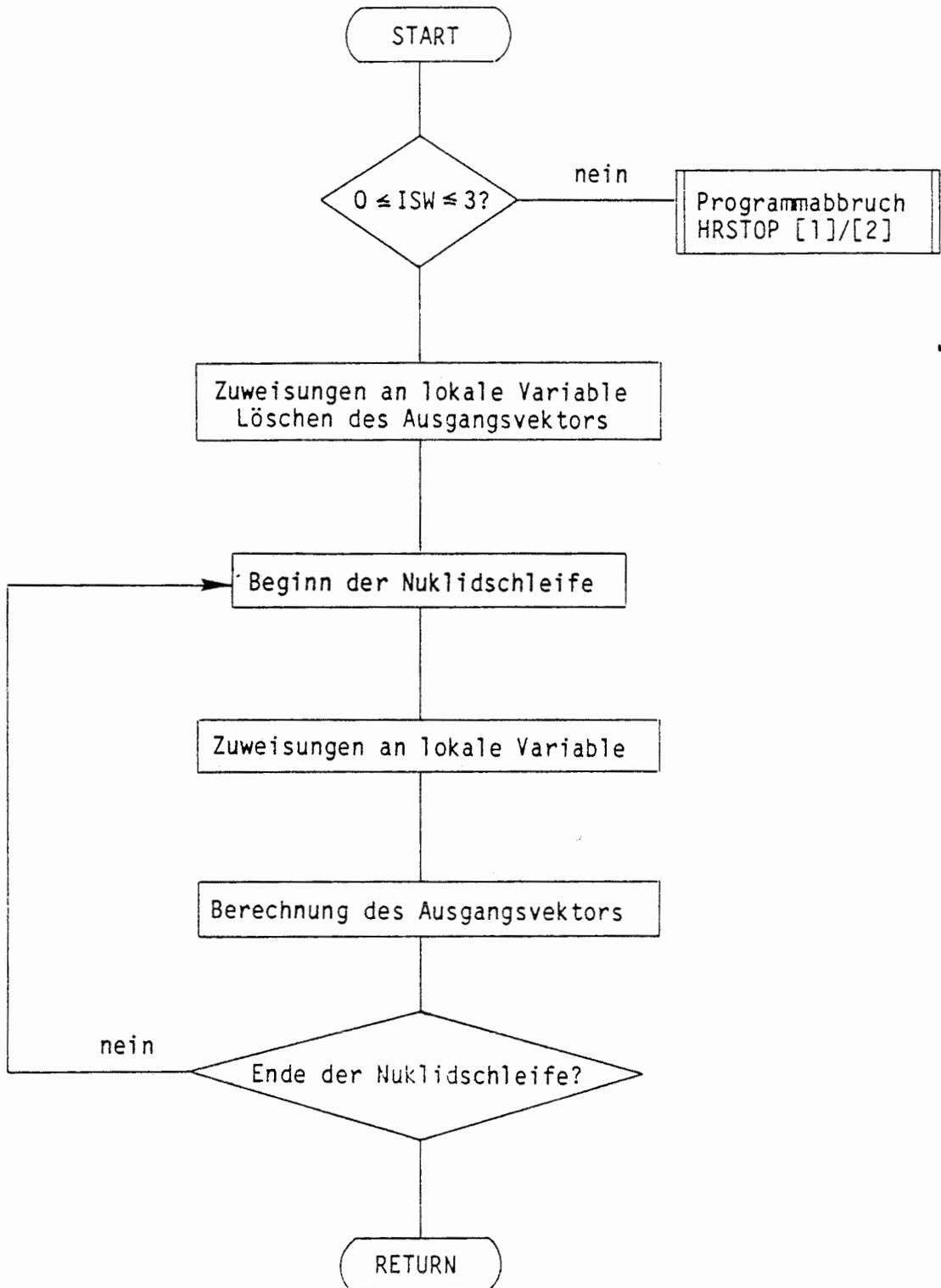
LESENDER ZUGRIFF AUF COMMON-BLÖCKE:

/WZNY/
/WZEY/ : NEY

UNTERPROGRAMMAUFRUFE:

[1] : HRSTOP('ELMAKO')
[2] : HRSTOP('NUMAKO')

ABLAUF DES PROGRAMMS ELMAKO/NUMAKO



LESENDER ZUGRIFF AUF COMMON-BLÖCKE:

/YC/ : CBANA

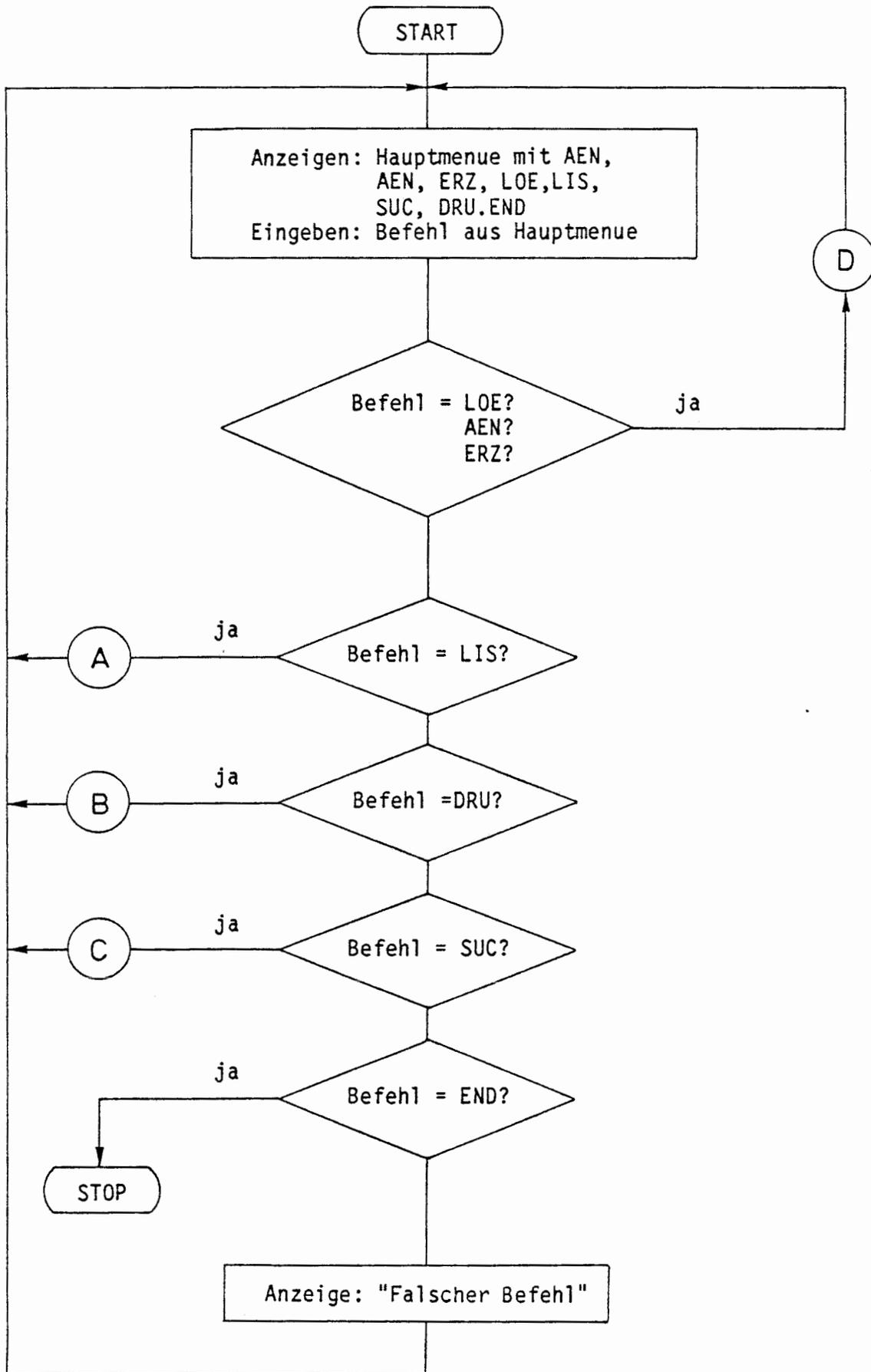
/YZ/ : DTMA

SCHREIBENDER ZUGRIFF AUF COMMON-BLÖCKE:

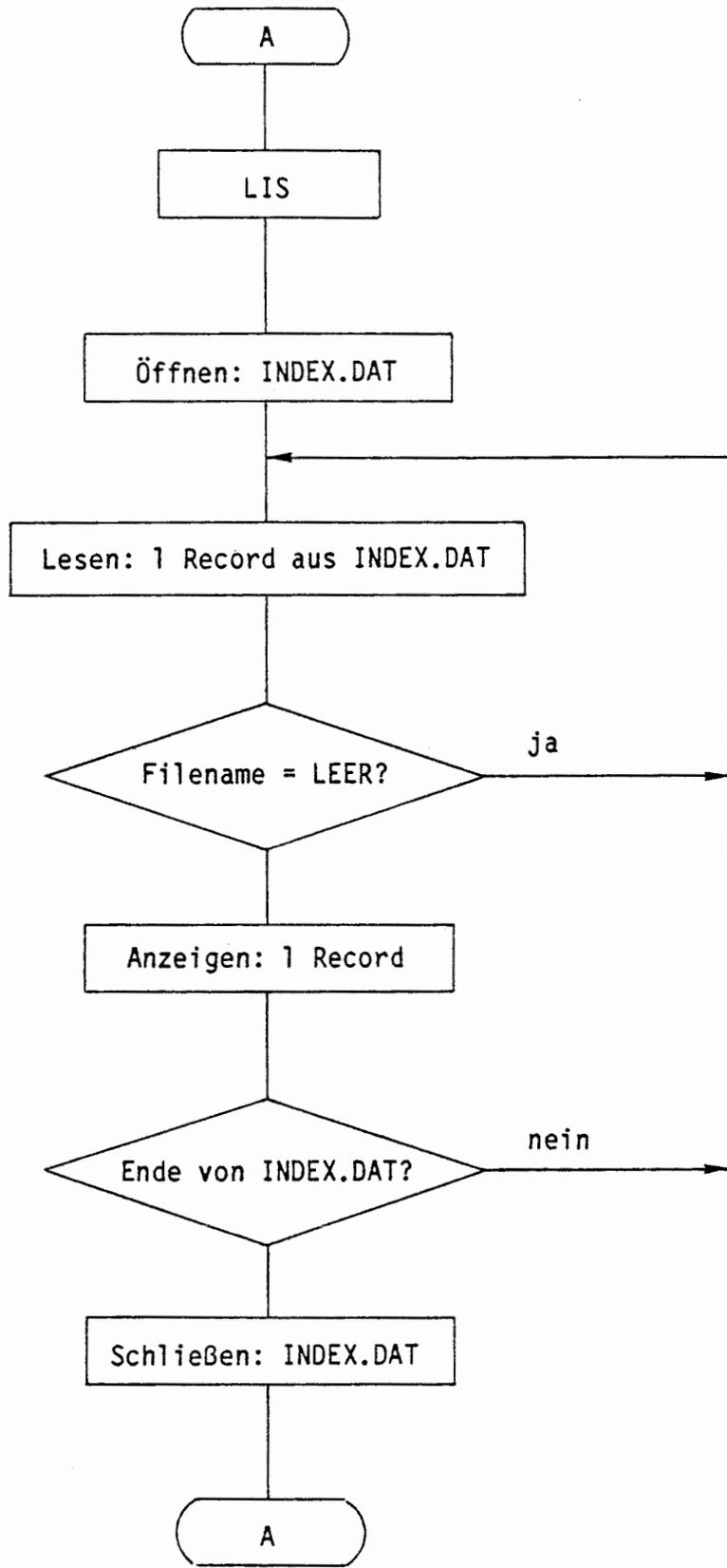
/YC/ : CBANAA, CNEFF

/YZ/ : DTMA

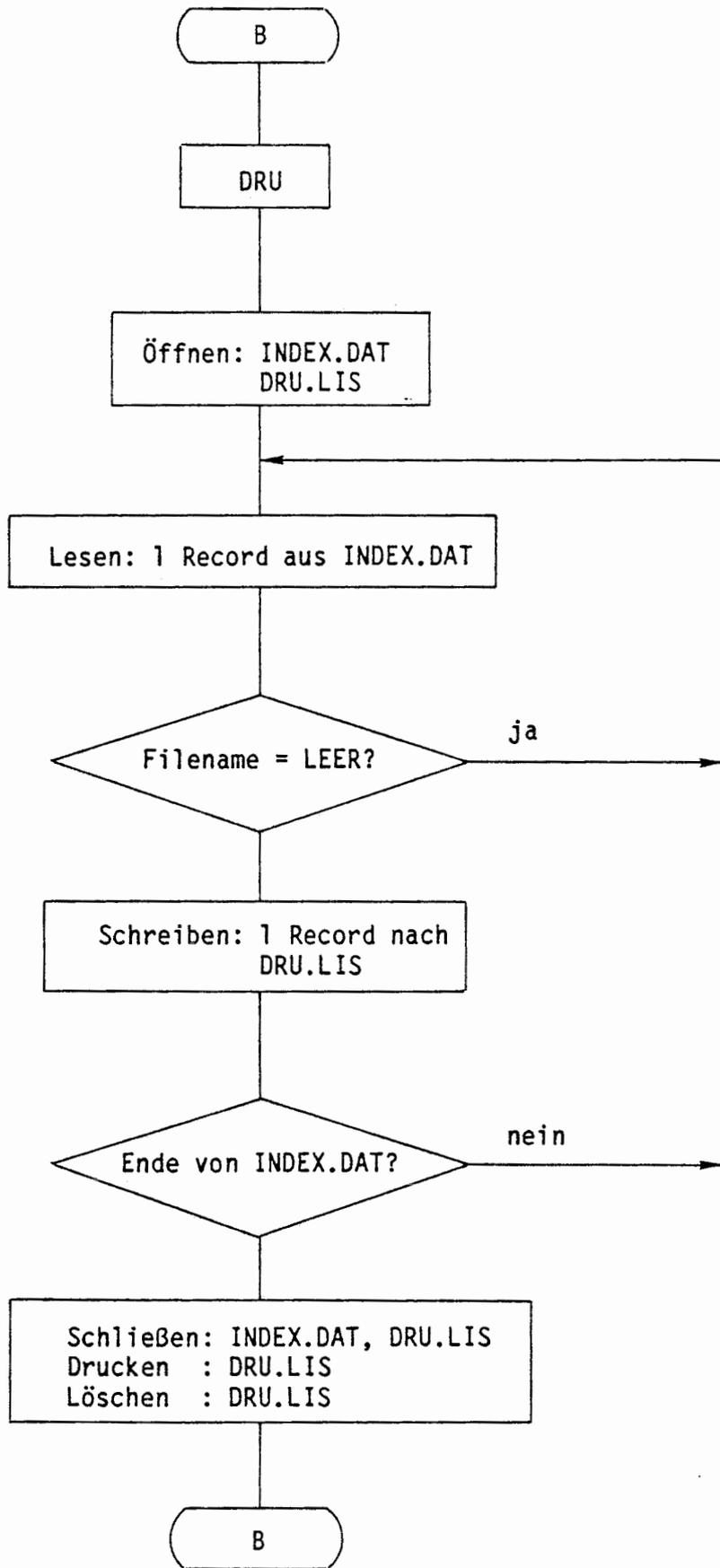
ABLAUF DER PROZEDUR JUNK



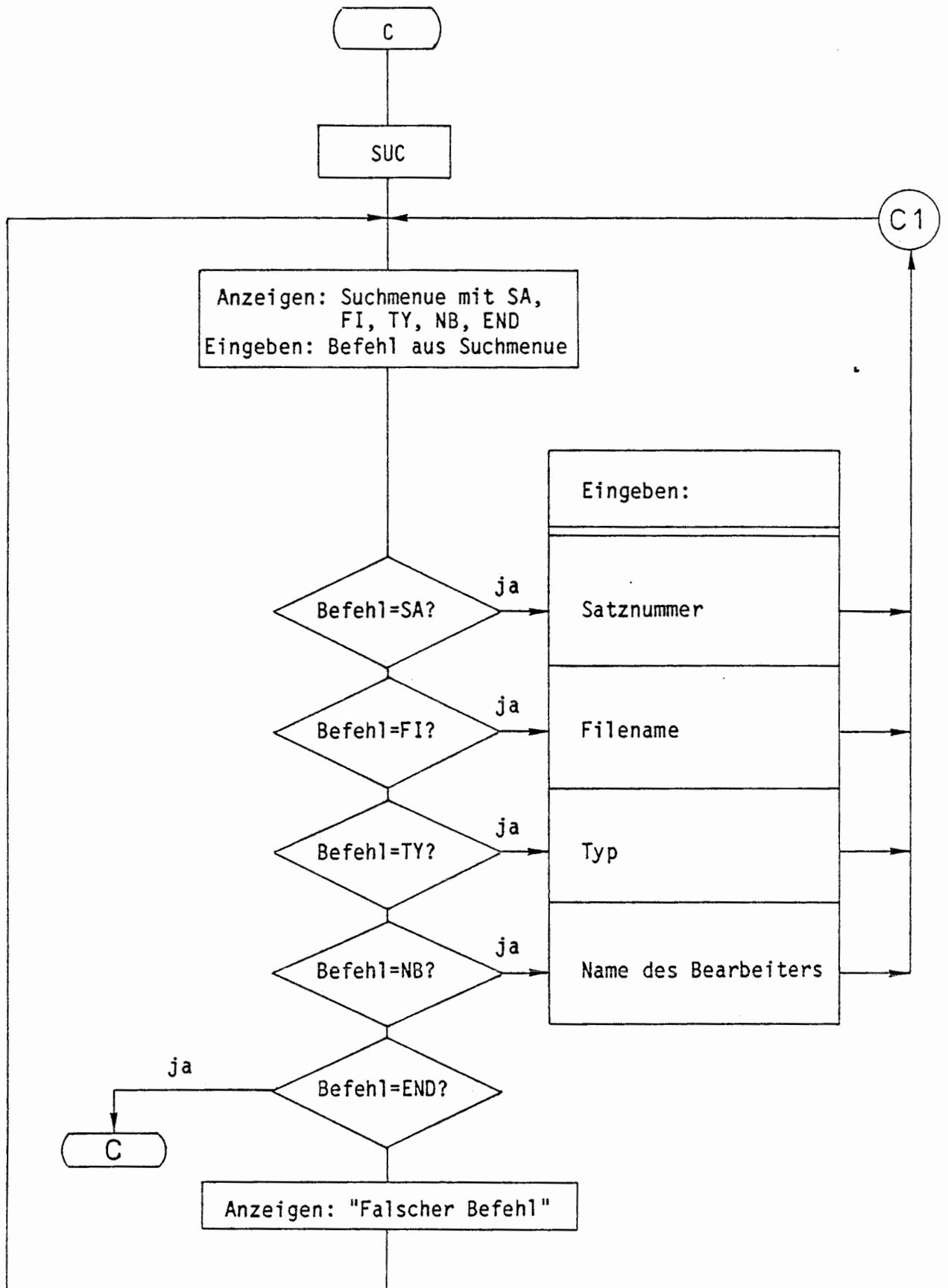
ABLAUF DER PROZEDUR JUNK TEILAUFGABE A

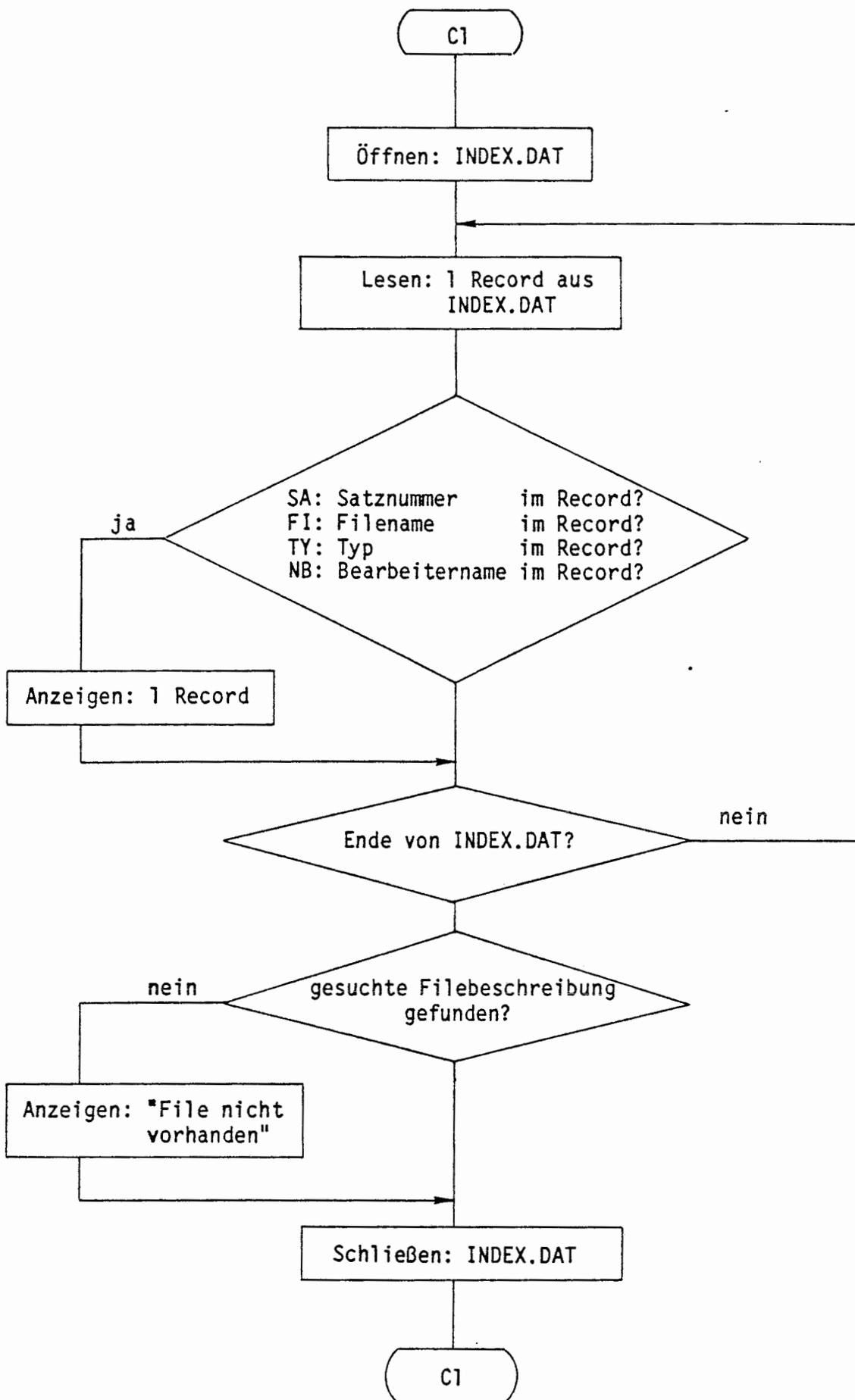


ABLAUF DER PROZEDUR JUNK TEILAUFGABE B

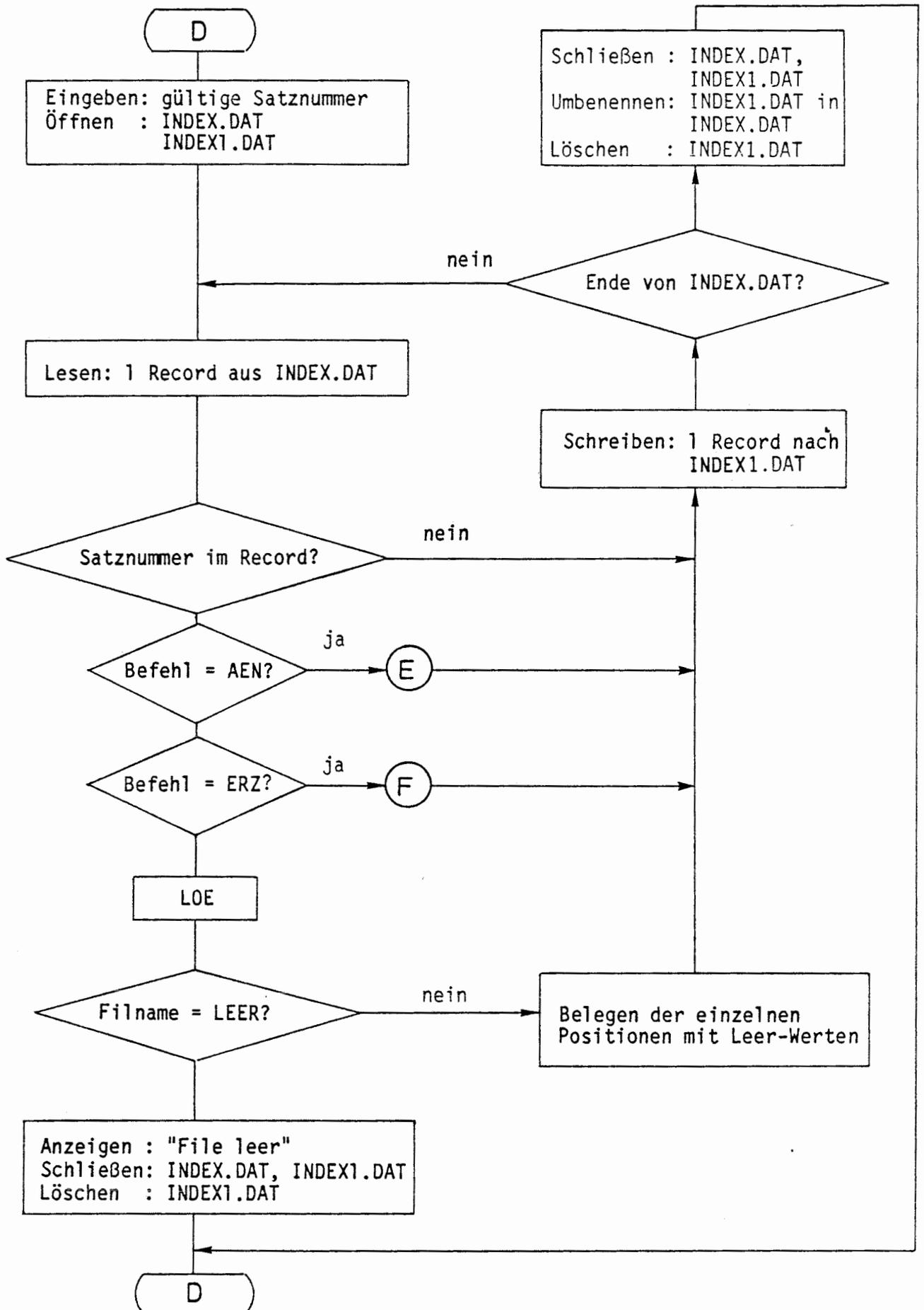


ABLAUF DER PROZEDUR JUNK TEILAUFGABE C

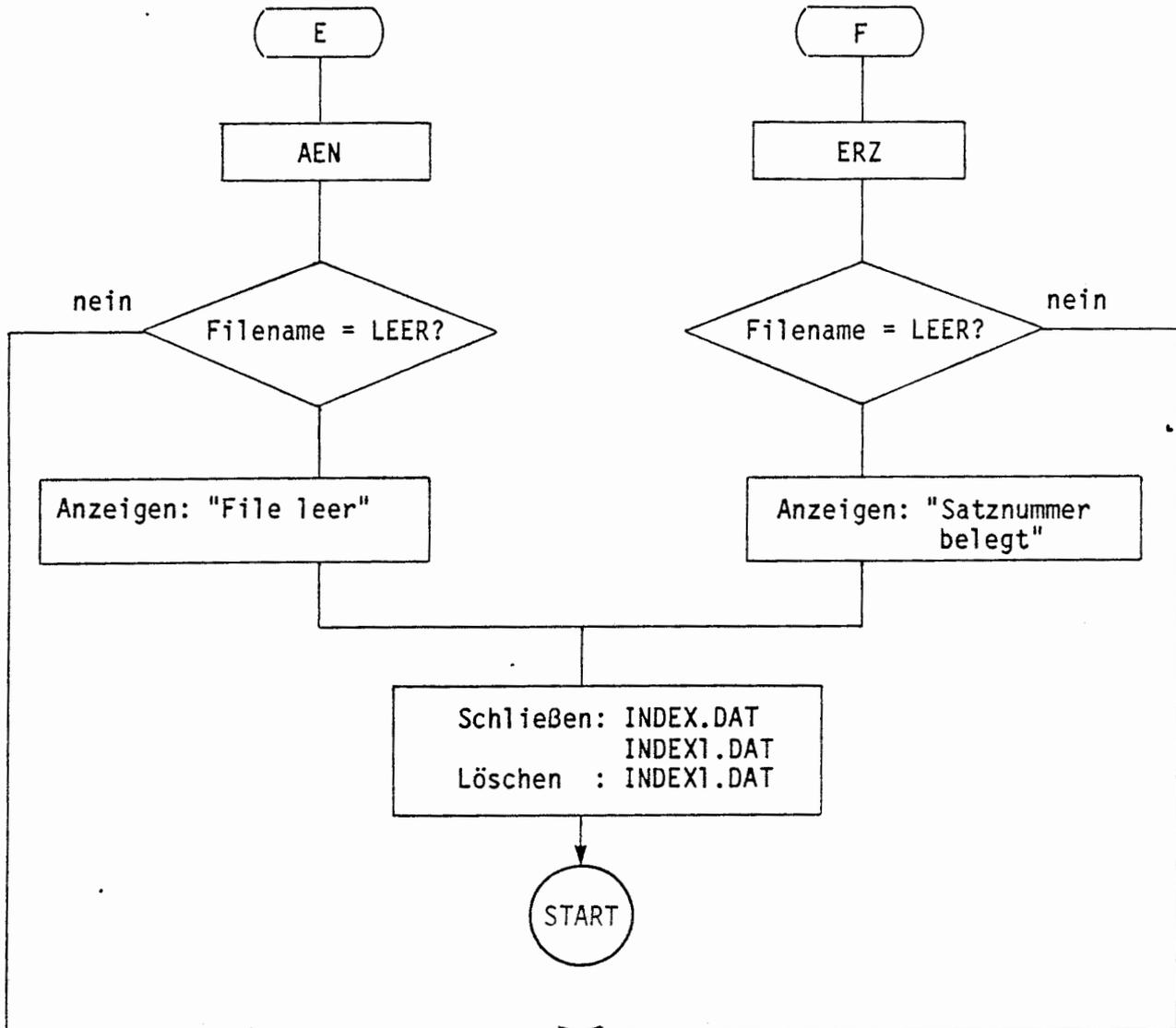




ABLAUF DER PROZEDUR JUNK TEILAUFGABE D



ABLAUF DER PROZEDUR JUNK TEILAUFGABE E UND F



Eingeben: - neuer Filname
- neuer Inhalt
- neuer Typ
- neuer Bearbeitername

Hinzufügen von Zusatzinformationen durch das System:
- aktuelles Datum und Zeit
- Blocklänge des neuen bzw. geänderten Files

E/F

A 2.5.6 AUSGABE WÄHREND DER RECHNUNG

a. Kontrollausgaben der Nuklid- und anderer Ströme

Vor und nach dem Aufruf eines Barrierenmodells innerhalb der Barrierenschleife kann durch setzen des Parameters $IDR(6) > 0$ im Job-Input-File eine Kontrollausgabe der Nuklid- und anderer Ströme erfolgen. Hieraus können Änderungen der relevanten Größen bei der Abarbeitung eines Barrierenmodells und der dort berechneten Effekte kontrolliert werden. Im Detail werden die Komponenten der folgenden Übergangsmatrizen ausgegeben (KBY: Nummer der Barriere):

RB3Y(K, KBY) = DV1(K):	geometrische Barrierendaten
RA1Y(K, KBY) = ZAN(K):	Aktivitätsinventar der Barriere
RA2Y(K, KBY) = ZAG(K):	Matrix-, Behältermasse, Gebinde-, Hohlraumvolumen, Zwischenlagerzeit
EVN(K) :	Eingangsvektor für Nuklidströme
EVS(K) :	Eingangsvektor für andere Ströme
RU1Y(K, KBY) = ZUN(K):	Nuklidübergangsströme
RU2Y(K, KBY) = ZUS(K):	Übergabedaten Ströme
RU3Y(K, KBY) = ZUD(K):	sonstige Übergabedaten

Die Ausgabe erfolgt für alle Zeitschritte, alle Barrieren und alle Nuklide und benötigt daher sehr viel Speicherplatz.

b. Ausgabe zeitabhängiger Größen: (vgl. Teilaufgabe D des Ablaufdiagramms)

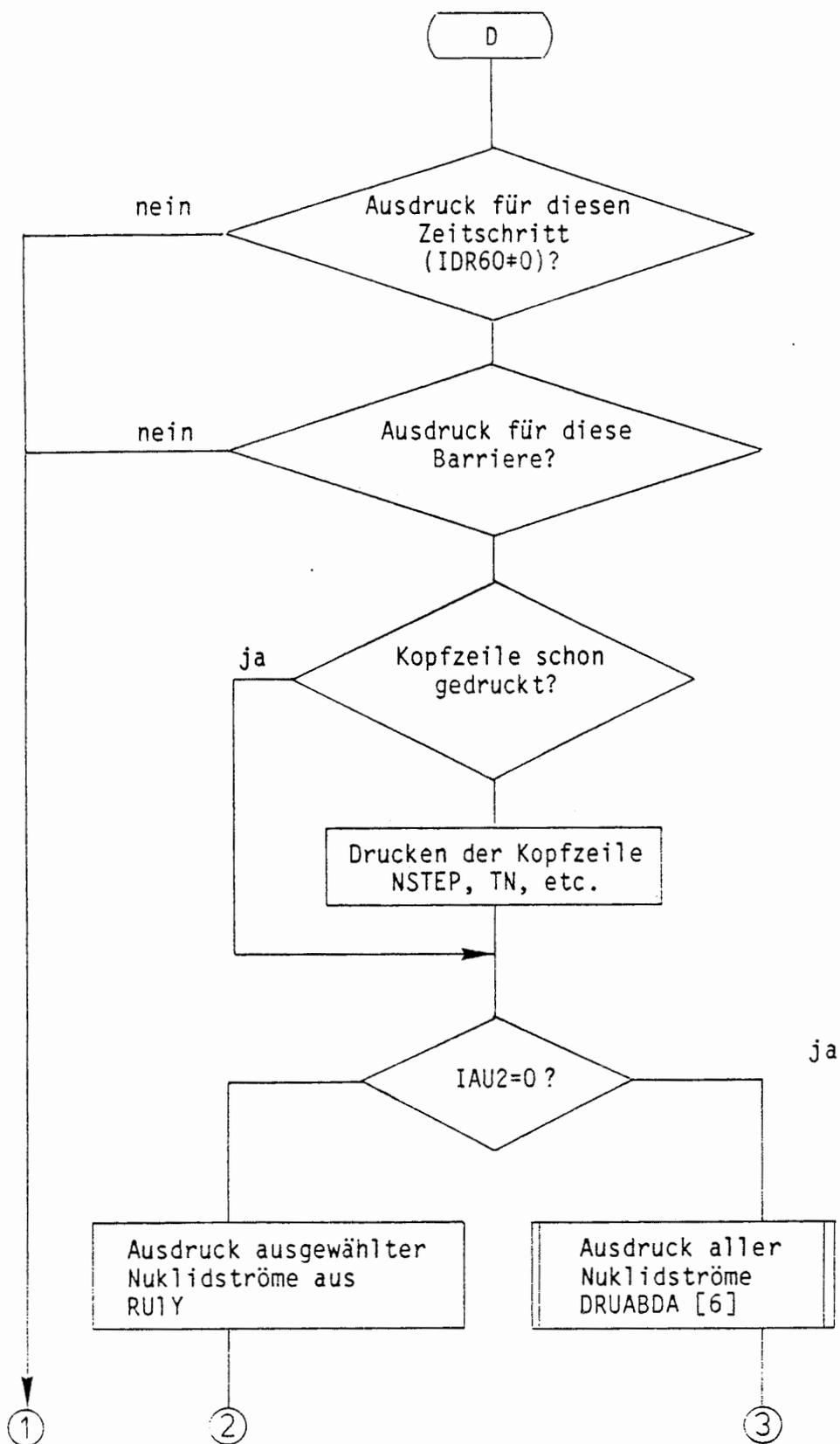
Allgemein erfolgt ein Ausdruck zeitabhängiger Größen während der Rechnung nur, falls der Steuerparameter $IDR60 > 0$ ist. Dieser Parameter wird zuvor (vgl. Teilaufgabe B des Ablaufdiagramms) aus den Parametern $IDR(1)$, $IDR(2)$ und den ausgewählten Zeitbereichen ermittelt (Job-Input-File: DRUCK). Für $IDR60 > 1$ erfolgt auch der Ausdruck für kumulierte Größen. Für $IAU(1)$ ausgewählte Barrieren (Auswahl im Job-Input-File) werden folgende Daten während der Rechnung (innerhalb der Barrierenschleife) ausgegeben.

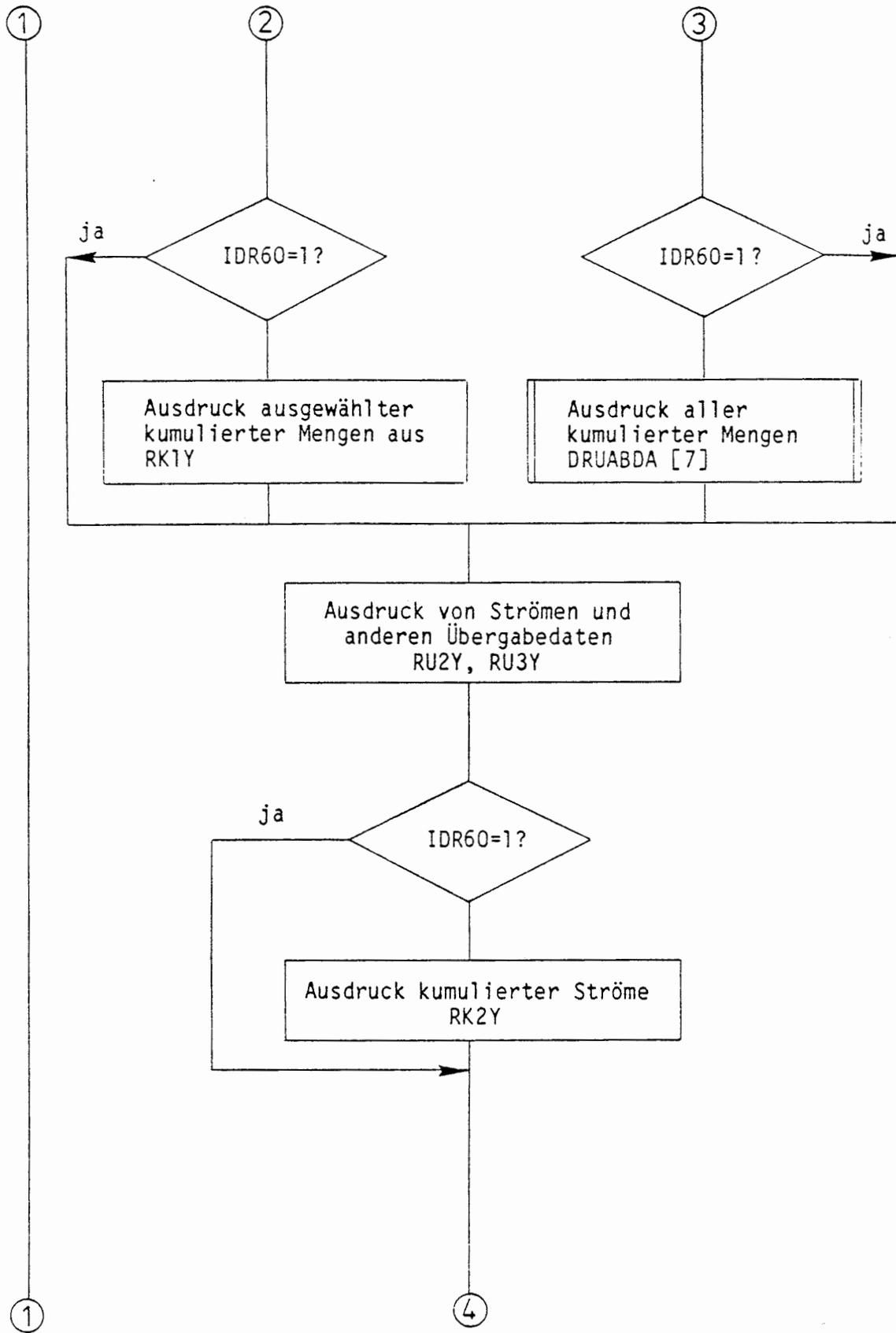
Kopfzeile: Zeitschritt NSTEP, aktuelle Zeit TN, Schrittweite DT, maximale Schrittweite DTMA, Name des Effekts, in dem DTMA festgelegt wurde, relative Stromänderung EPS, Nuklidname und Atomgewicht bzw. Name und Nummer des Übergangstroms, durch den EPS festgelegt wurde. Hiernach erfolgt der Ausdruck ausgewählter Aktivitätsströme (RU1Y) bzw. ausgewählter kumulierter Mengen (RK1Y), falls IAU(2)>0 ist, oder der Ausdruck aller Nuklidströme bzw. aller kumulierter Mengen durch Aufruf des Unterprogramms DRUABDA, falls IAU(2)=0 ist. Danach erfolgt der Ausdruck von Strömen (RU2Y) und anderen Übergabedaten (RU3Y) bzw. der Ausdruck kumulierter Ströme.

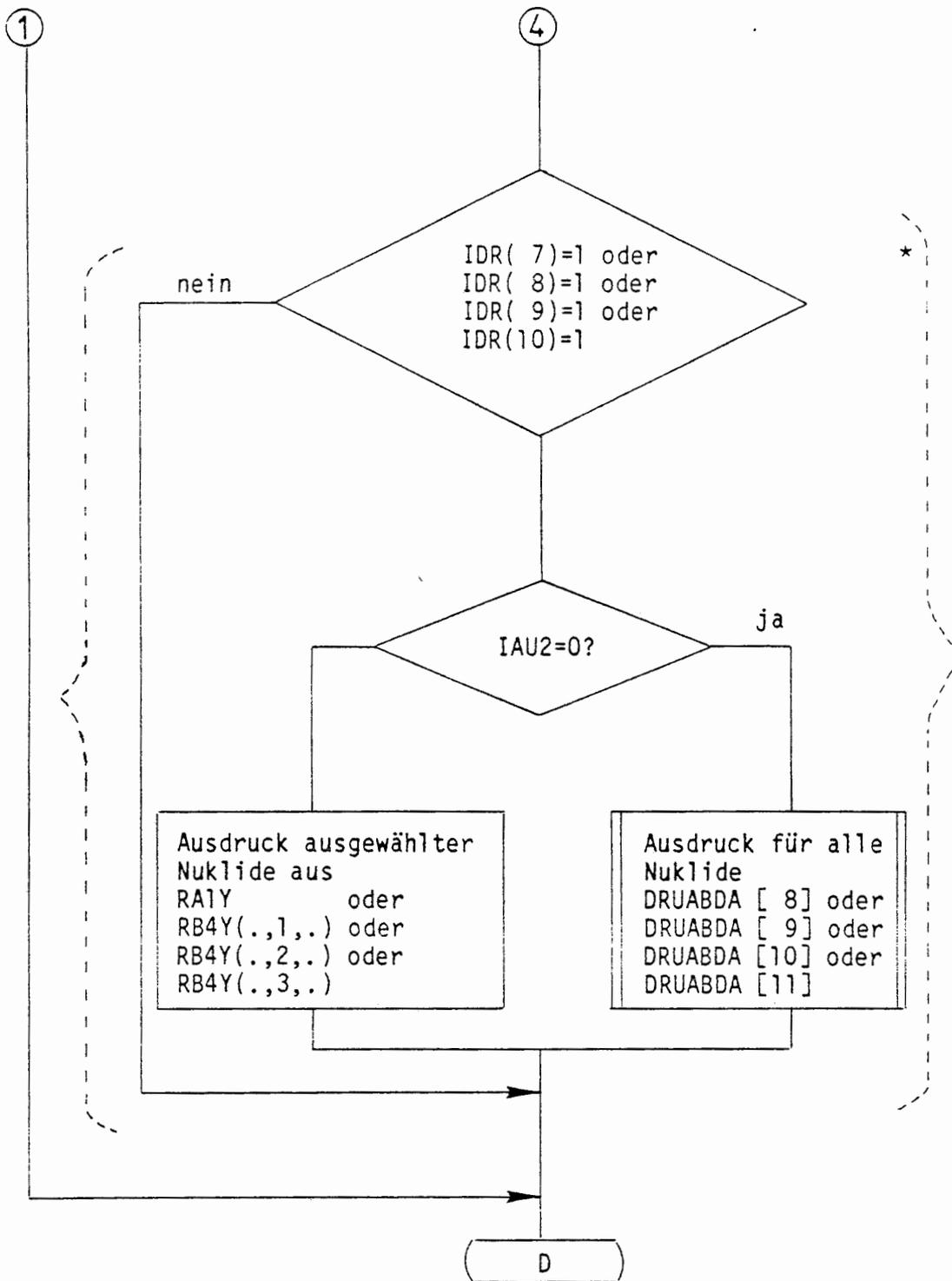
Die Parameter IDR(7) bis IDR(10) des Job-Input-Files (Druck) steuern den Ausdruck weiterer nuklidspezifischer Daten. Ist der Auswahlparameter IAU(2)=0, erfolgt stets der Ausdruck für alle Nuklide mit Hilfe des Unterprogramms DRUABDA. Ist IAU(2) > 0, erfolgt der Ausdruck für IAU(2) ausgewählte Nuklide.

IDR(7)=1 : Ausdruck des Aktivitätsinventars der ausgewählten Barrieren
IDR(8) bis IDR(10)=1 : Ausdruck (barrierenspezifischer) Nuklid-
datensätze

ABLAUF DES PROGRAMMS STEBA TEILAUFGABE D







* Der Bereich in den gestrichelten Klammern tritt viermal auf mit jeweils einem der angegebenen Ausdrücke.

A 2.5.7 AUSGABE AM ENDE DER RECHNUNG

Beim letzten Durchlauf der Zeitschleife ($TN > DASZ$) wird der Steuerparameter $IDR60 > 0$ gesetzt, wenn $IDR2 \neq 0$ (vgl. Teilaufgabe B im Ablaufdiagramm), somit erfolgt in jedem Fall nach Beendigung der Rechnung ein Ausdruck der Ergebnisse gemäß A 2.5.6 Abschnitt b.

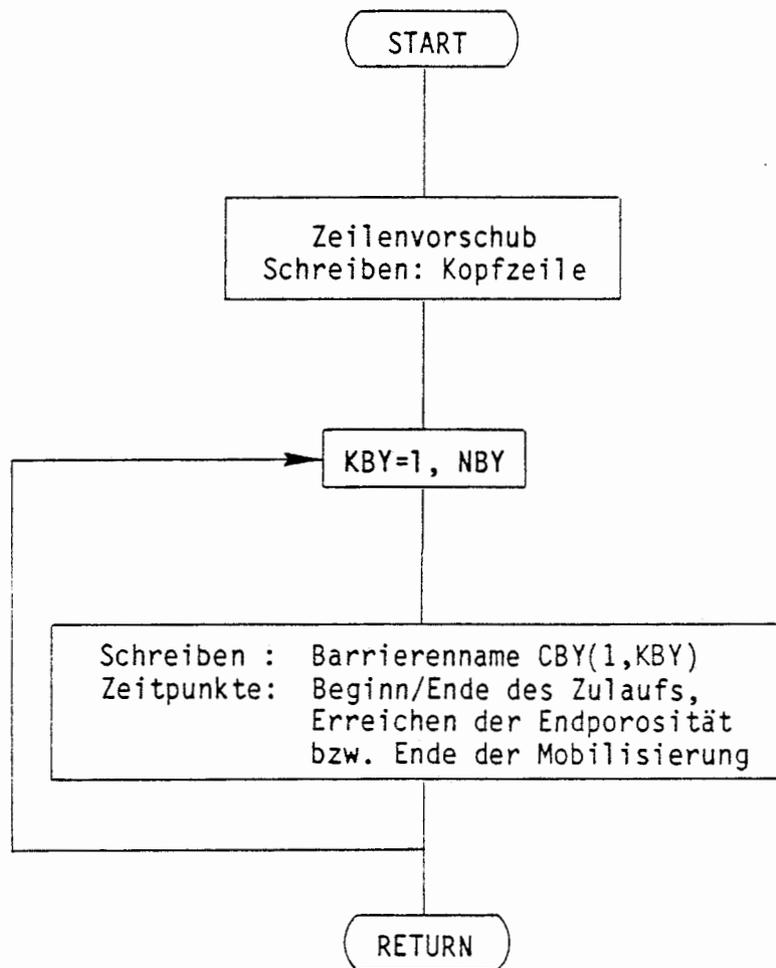
Wird im Rahmen eines Szenarios der Lösungszufluß modelliert, so kann man einen Überblick über den Verlauf des Lösungszuflusses in das Grubengebäude gewinnen, wenn die Lösungszuflußzeiten in die einzelnen Barrieren bekannt sind. Hierzu wird am Ende der Rechnung eine Tabelle ausgegeben, die für jede Barriere die Zeitpunkte des Beginns und des Endes des Lösungszuflusses enthält. Weiterhin entnimmt man der Tabelle das Ende der Mobilisierung bzw. den Zeitpunkt des Erreichens der Endporosität, wenn eine konvergenzbedingte Porositätsabnahme modelliert wird.

Zusätzlich können durch Steuerparameter IDR im Job-Input-File (DRUCK) sowie durch Steuerparameter IOU bzw. IME im Job-Input-File (OUTPUT bzw. MENGEN) weitere Ausgaben am Ende der Rechnung bewirkt werden.

A 2.5.7.1 AUSGABE DES ZUFLUSSVERLAUFS

Der Zuflußverlauf in das Grubengebäude kann durch Ausgabe der Zeitpunkte Beginn des Laugenzuflusses, Ende des Laugenzuflusses und Erreichen der Endporosität bzw. Ende der Mobilisierung für jede Barriere verfolgt werden. Die Ausgabe einer Tabelle dieser Zeitpunkte wird durch das Unterprogramm ZEITEN durchgeführt.

ABLAUF DES PROGRAMMS ZEITEN



A 2.5.7.2 AUSGABE VON TABELLEN MIT LÖSLICHKEITSZEITPUNKTEN UND MIT
MAXIMALEN ELEMENTKONZENTRATIONEN

Durch Setzen des Steuerparameters IDR(3)=1 im Job-Input-File (DRUCK) werden am Ende der Rechnung Aufrufe der Unterprogramme PRTLГ (ENTRY von SETLG) und PRTEL (ENTRY von MAXEL) zum Ausdruck einer Tabelle (lokales Feld in SETLG) über Zeitpunkte des erstmaligen Erreichens der Löslichkeitsgrenzen und einer Tabelle (lokales Feld in MAXEL) mit maximalen Elementkonzentrationen bewirkt.

Der Aufruf des Unterprogramms SETLG im Programm STEBA zu Beginn der Rechnung sorgt für eine Vorbesetzung aller Löslichkeitszeitpunkte mit 0. Das Eintragen der Löslichkeitszeitpunkte in die Tabelle wird durch Aufruf von SETLG von Unterprogrammen für Effekte vorgenommen.

Aufruf des Unterprogramms MAXEL im Programm STEBA zu Beginn der Rechnung sorgt für eine Vorbesetzung aller auszugebenden Größen mit 0. Das Eintragen der Elementkonzentration (hier ist eine hypothetische Konzentration, nämlich das Gesamtinventar eines Elements dividiert durch das Lösungsvolumen, gemeint) zum Zeitpunkt der größten Konzentration, des Zeitpunktes, der Nummer des Zeitschritts, der Löslichkeitsgrenze, der Kapazitätsgrenze, des KD-Wertes, des gelösten Anteils und dessen Konzentration in die Tabelle erfolgt bei Aufruf von MAXEL von Unterprogrammen für Effekte.

ÜBERSICHT ÜBER DAS PROGRAMM SETLG, ENTRY PRTLГ

ÜBERGEORDNETES PROGRAMM :	STEBA
LÄNGE DES PROGRAMMS :	133 Zeilen 557 Bytes
STAND DER DOKUMENTATION :	15. März 1986

KURZBESCHREIBUNG:

Erstellen und Ausdrucken einer Tabelle (lokales Feld) mit den Zeitpunkten des erstmaligen Erreichens der Löslichkeitsgrenzen in jeder Barriere

PARAMETERLISTE:

NUMEL

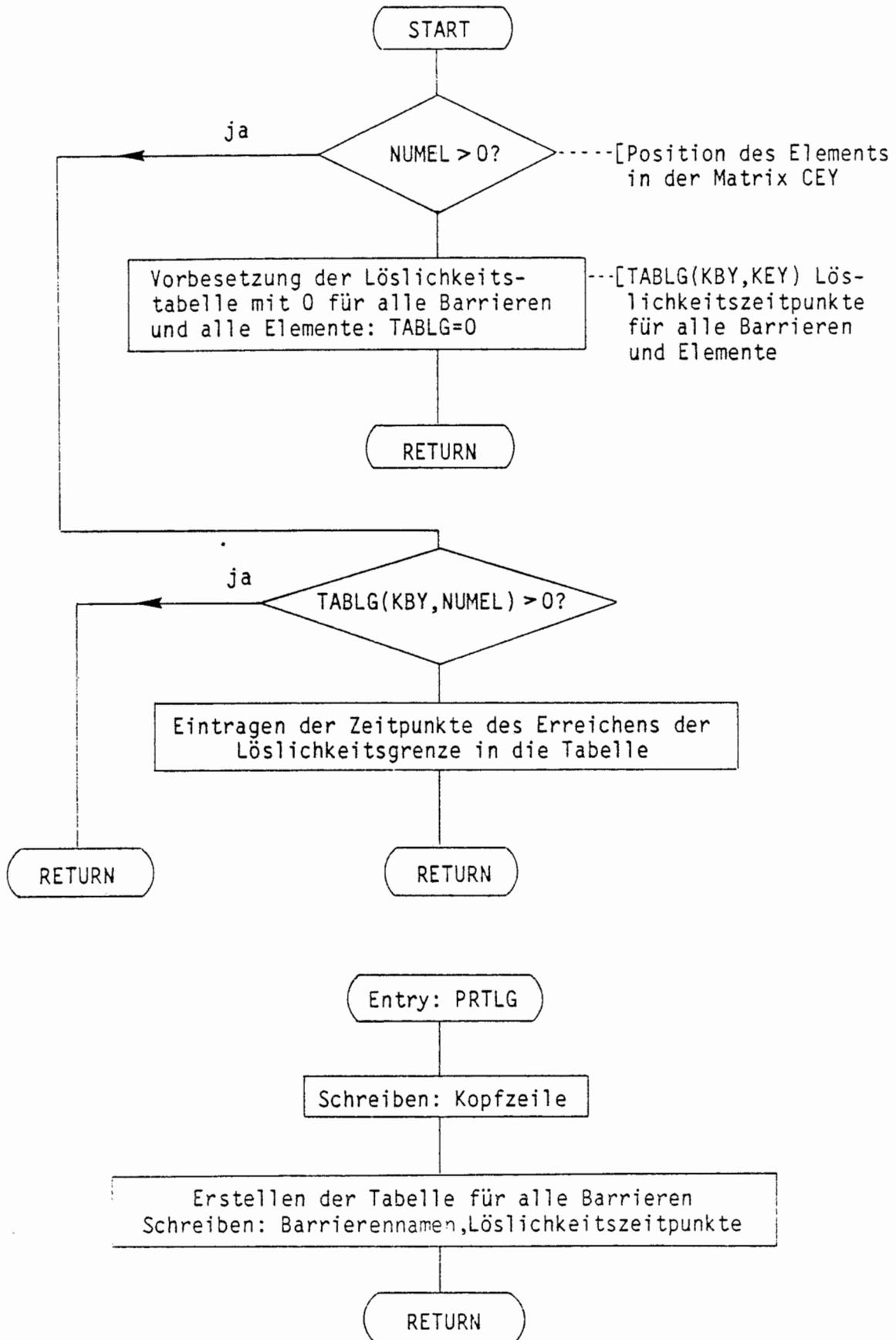
EINGANGSPARAMETER:

NUMEL : Position eines Elements in der Matrix CEY

LESENDER ZUGRIFF AUF COMMON-BLÖCKE:

/YZ/ : TN, KBY
/WCEY/ : CEY
/WZEY/ : NEY
/WCBY/ : CBY
/WZBY/ : IB1Y, NBY
/WTEXT/ : CBLANK

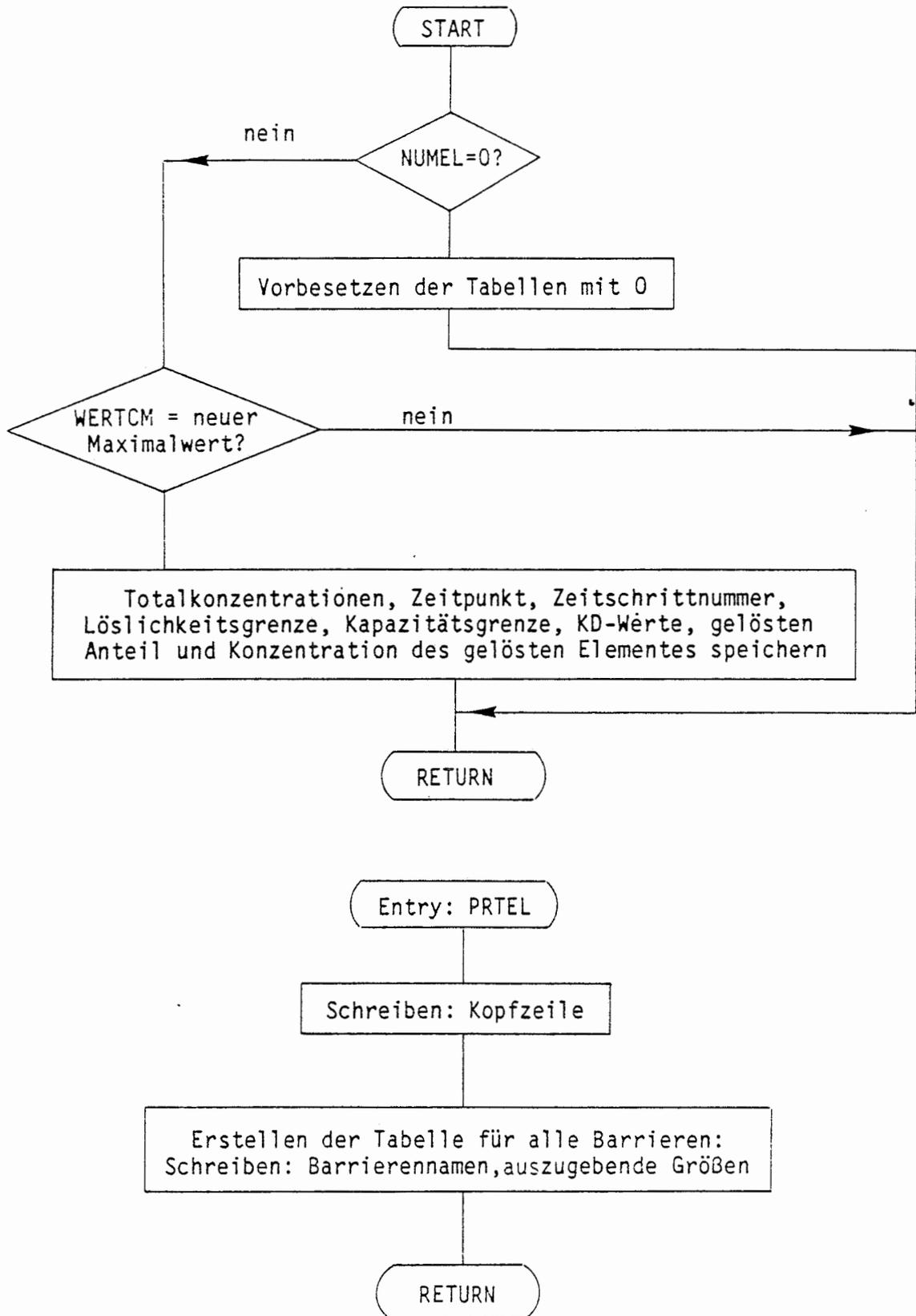
ABLAUF DES PROGRAMMS SETLG/ENTRY PRTL



LESENDER ZUGRIFF AUF COMMON-BLÖCKE:

/YZ/ : TN, NSTEP, KBY
/WCEY/ : CEY
/WZEY/ : NEY
/WCBY/ : CBY
/WZBY/ : IB1Y, NBY

ABLAUF DES PROGRAMMS MAXEL/ENTRY PRTTEL



A 2.5.7.3 AUSGABE VON FREISETZUNGSMENGEN

- a. Ergebnisausgabe entsprechend den Parametern IOU des Job-Input-Files (OUTPUT) (vgl. Teilaufgabe G im Ablaufdiagramm)

Für jede der folgenden Ausgaben wird mit Hilfe des Funktionsunterprogramms NOUTBAR geprüft, ob für diese Barriere (Nummer KBY), ein Ergebnisausdruck mit Hilfe des Unterprogramms DRUABDA erfolgen soll.

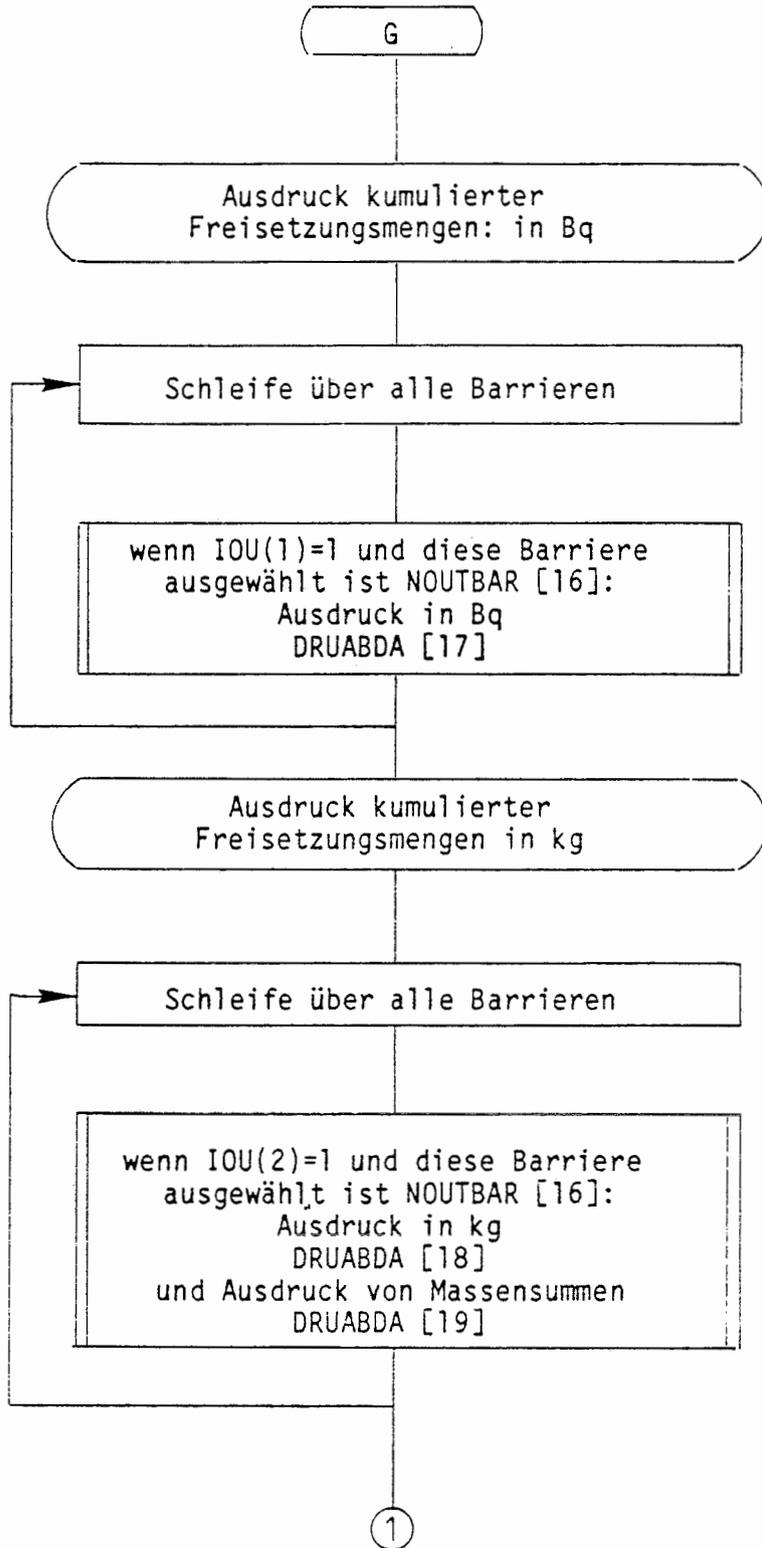
Ist $IOU(1) > 0$, so werden kumulierte Freisetzungsmengen in Bq ausgedruckt, ist $IOU(2) > 0$, so werden kumulierte Freisetzungsmengen in kg und ausgewählte Massensummen ausgedruckt. Für $IOU(3) > 0$ erfolgt der Ausdruck kumulierter relativer Freisetzungsmengen. Die auszugebenden Größen werden jeweils aus den kumulierten Mengen für Nuklide RK1Y ermittelt.

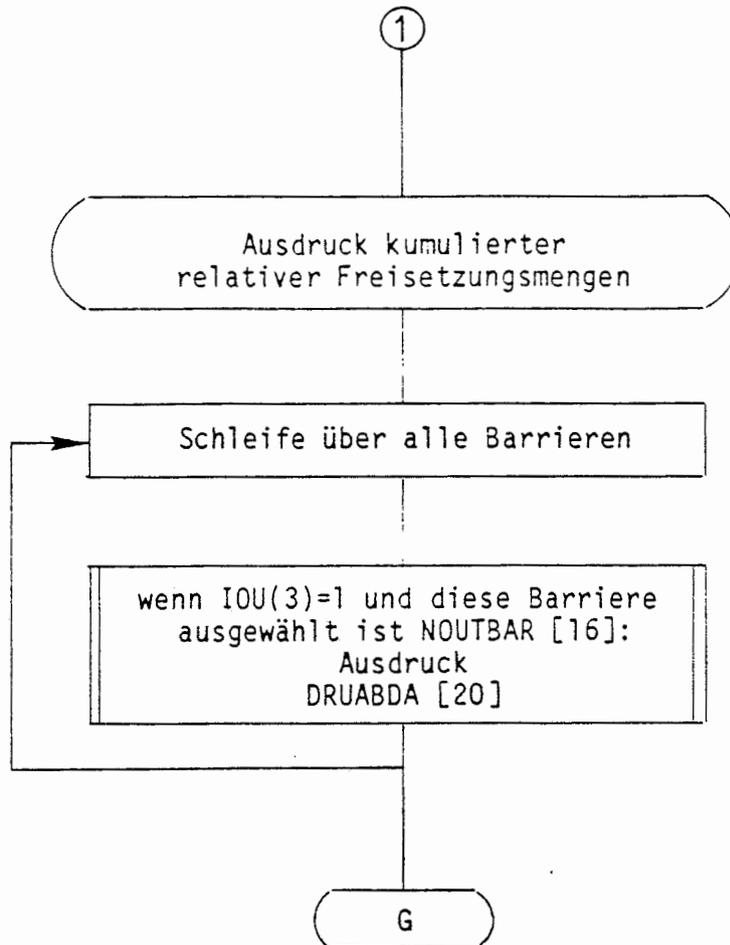
- b. Ausdruck von Freisetzungsmengen und -raten entsprechend den Parametern IME des Job-Input-Files (MENGEN)

Für im Job-Input-File (MENGEN) ausgewählte Barrieren und Zeitbereiche werden am Ende der Rechnung folgende Größen zusätzlich mit Hilfe des Unterprogramms DRUFM ausgegeben.

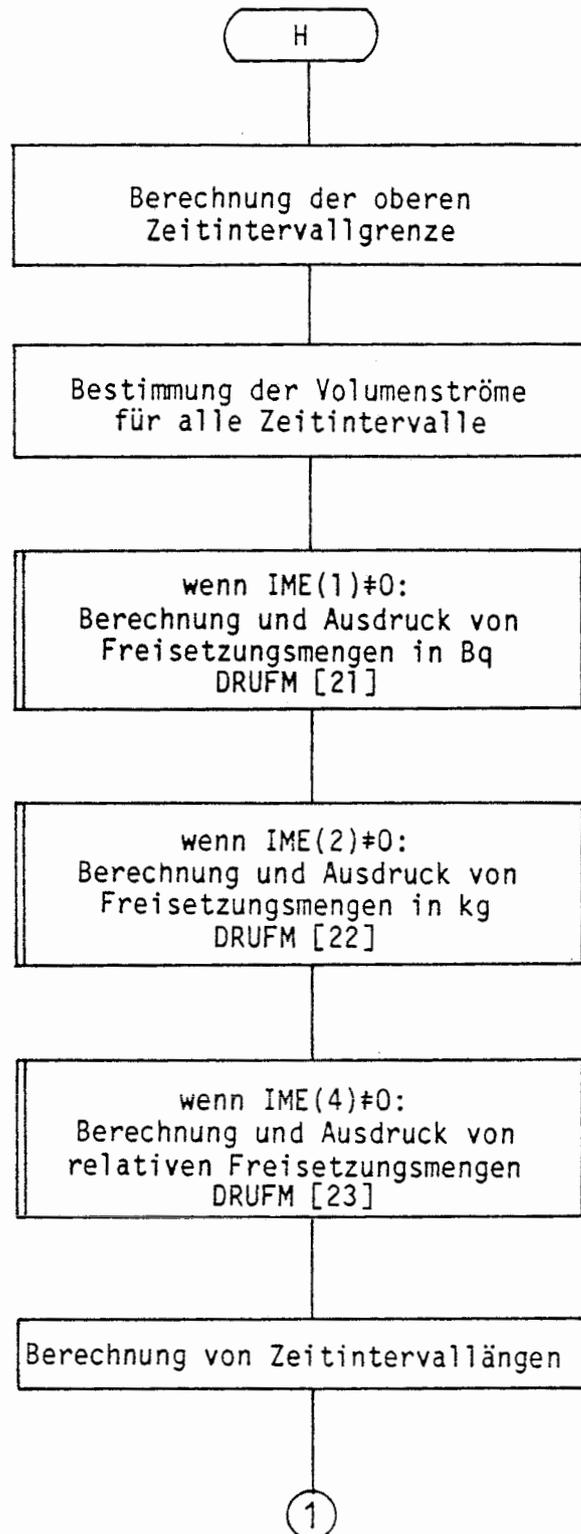
Ist $IME(1) > 0$, so werden Freisetzungsmengen in Bq für jedes Nuklid ausgegeben, für $IME(2) > 0$ erfolgt die Ausgabe für Mengen in kg. Bei $IME(4) > 0$ werden relative Freisetzungsmengen für jedes Nuklid ausgedruckt. $IME(6)=1$ bzw. $IME(7)=1$ bewirken die Ausgabe von Freisetzungsraten in Bq/a bzw. kg/a, $IME(6)>1$ bzw. $IME(7)>1$ bewirken die Ausgabe von Freisetzungsraten in Bq/s bzw. kg/s. Die auszugebenden Größen werden jeweils aus den Freisetzungsraten der Nuklide RF2Y in den Zeitintervallen ZEFT ermittelt (vgl. Teilaufgabe H im Ablaufdiagramm).

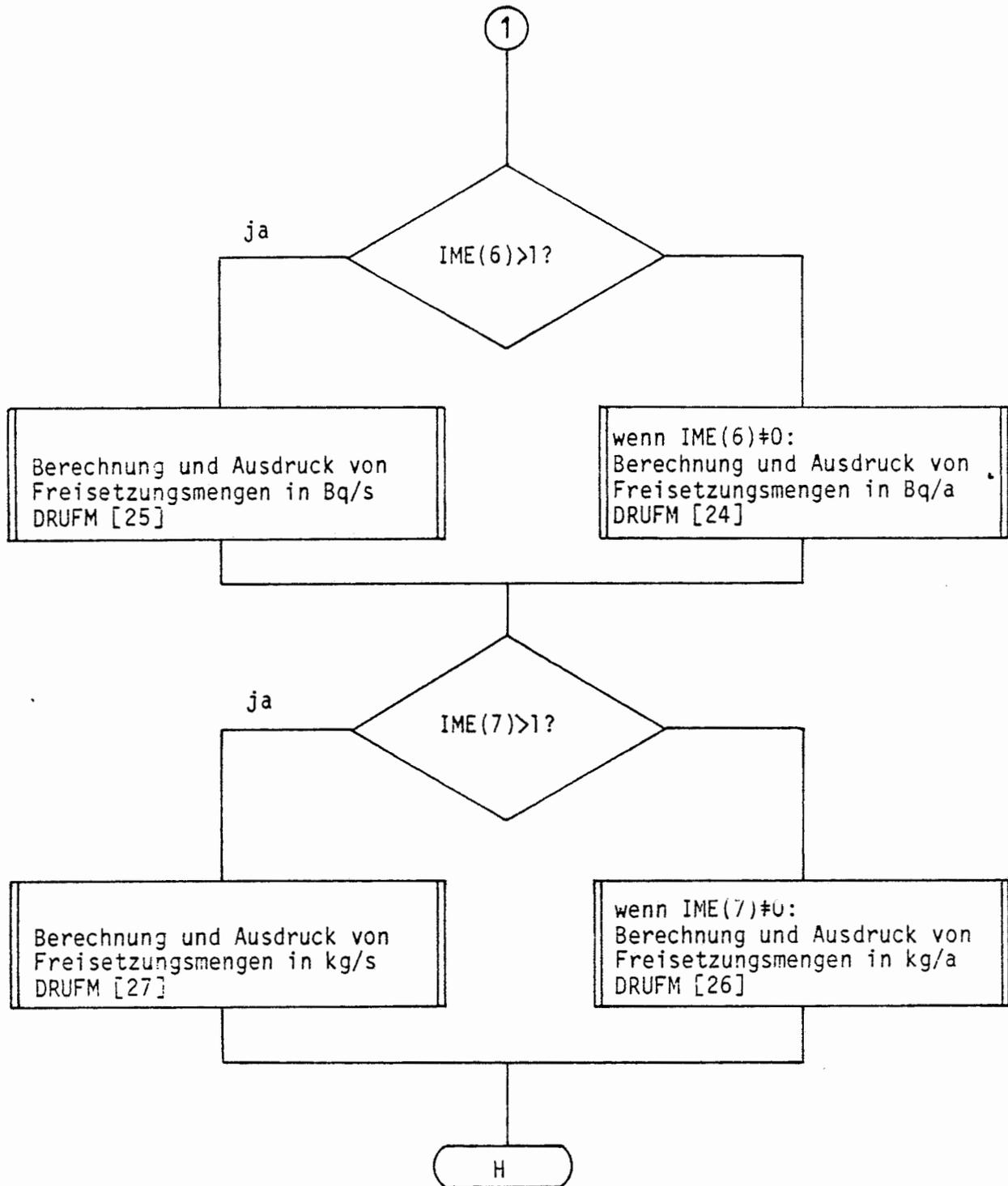
ABLAUF DES PROGRAMMS STEBA TEILAUFGABE G





ABLAUF DES PROGRAMMS STEBA TEILAUFGABE H





ÜBERSICHT ÜBER DAS PROGRAMM DRUFM

ÜBERGEORDNETES PROGRAMM : STEBA

LÄNGE DES PROGRAMMS : 135 Zeilen
814 Bytes

STAND DER DOKUMENTATION : 15. März 1986

KURZBESCHREIBUNG:

Ausdruck einer Tabelle für Freisetzungsmengen oder Freisetzungsraten für 5 Zeitintervalle

PARAMETERLISTE:

RKUM1(LEY, LNY), TFO, IP, CBANA, ZEFT(5)

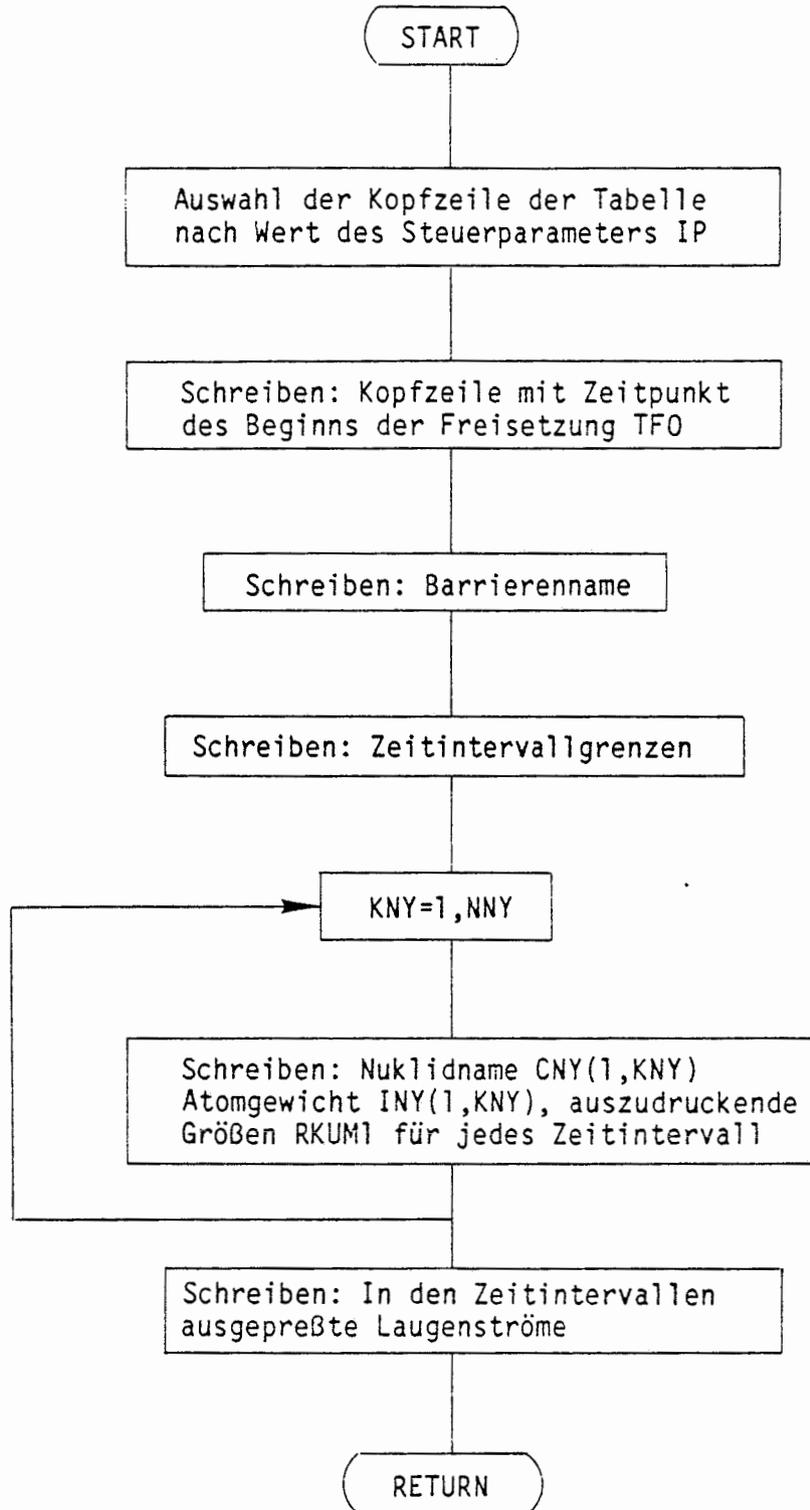
EINGANGSPARAMETER:

RKUM1 : auszudruckende Größe
TFO : Beginn der Freisetzung
IP : Steuerparameter für Kopfzeile der Tabelle
CBANA : Barrierenname
ZEFT : Zeitintervallgrenzen

LESENDER ZUGRIFF AUF COMMON-BLÖCKE:

/WCNY/, /WZNY/, /WZFY/

ABLAUF DES PROGRAMMS DRUFM



16

A 3 COMMON - BLOECKE

Inhalt dieses Kapitels ist die Beschreibung der COMMON-Bloেকে, die in dem Programm EMOS2 verwendet werden.

In einem ersten Unterkapitel A 3.1 werden einige Konventionen ueber die Namenswahl fuer die Variablen und Felder in COMMON-Bloেকে dargestellt. Eine Uebersicht ueber die Dimensionierung der Felder wird in Unterkapitel A 3.2 gegeben. Unterkapitel A 3.3 enthaelt einige Erlaeuterungen zu den Namen der COMMON-Bloেকে und ihre eigentliche Beschreibung.

Der Speicherbereich fuer eine bestimmte Gruppe von Eingangsdaten, beispielsweise barrierenspezifische Daten, bestand in einem frueheren Entwicklungsstand des Programmes lediglich aus einer Matrix, waehrend er nun in der Regel mehrere Matrizen umfasst. In der Kommentierung des Quelltextes wird noch immer von einer Matrix, z.B. der Barrierenmatrix, gesprochen, wenn eine dieser Matrizen gemeint ist.

A 3 COMMON - BLOECKE

Inhalt dieses Kapitels ist die Beschreibung der COMMON-Bloেকে, die in dem Programm EMOS2 verwendet werden.

In einem ersten Unterkapitel A 3.1 werden einige Konventionen ueber die Namenswahl fuer die Variablen und Felder in COMMON-Bloেকে dargestellt. Eine Uebersicht ueber die Dimensionierung der Felder wird in Unterkapitel A 3.2 gegeben. Unterkapitel A 3.3 enthaelt einige Erlaeuterungen zu den Namen der COMMON-Bloেকে und ihre eigentliche Beschreibung.

Der Speicherbereich fuer eine bestimmte Gruppe von Eingangsdaten, beispielsweise barrierenspezifische Daten, bestand in einem frueheren Entwicklungsstand des Programmes lediglich aus einer Matrix, waehrend er nun in der Regel mehrere Matrizen umfasst. In der Kommentierung des Quelltextes wird noch immer von einer Matrix, z.B. der Barrierenmatrix, gesprochen, wenn eine dieser Matrizen gemeint ist.

A 3.1 BEZEICHNUNG VON VARIABLEN UND FELDERN

Fuer die Namen der Variablen und Felder werden im folgenden einige Konventionen vereinbart, die der besseren Uebersicht und der Vermeidung von gleichen Namen fuer verschiedene Variablen dienen. Die dargestellten Regeln fuer die Namenswahl sind derzeit erst in Teilen des Programmes, insbesondere in und unterhalb von DALEKO (siehe Kap. A 2.4) realisiert.

1. Die Namen der Variablen und Felder beginnen mit einem bestimmten Buchstaben:

C	CHARACTER - Feld
I	INTEGER*4 - Feld
R	REAL*8 - Feld
L	Maximale Zahl der Komponenten in einer Dimension eines Feldes
N	Benutzte Zahl der Komponenten in einer Dimension eines Feldes
K	Zaehlvariable

2. Der zweite Buchstabe der Felder und der zweite oder dritte Buchstabe der INTEGER-Variablen kennzeichnet den inhaltlichen Bezug der Variablen bzw. Felder. Die verwendeten Buchstaben und ihre Bedeutung sind in der folgenden Tab. A 3-1 zusammengestellt.

Buchstabe	:	Inhaltlicher Zusammenhang
A	:	Matrizen mit abfallspezifischen Daten
B	:	Matrizen mit barrierenspezifischen Daten
E	:	Matrizen mit elementspezifischen Daten
N	:	Matrizen mit nuklidspezifischen Daten
S	:	Matrizen mit Strukturdaten
F	:	Matrizen mit Daten der freigesetzten Aktivitaeten
U	:	Matrizen mit Uebergangsdaten
I	:	Matrizen mit Barriereninventardaten
K	:	Matrizen mit Daten der kumulierten Mengen

Tab. A 3-1: Kennzeichnungen des inhaltlichen Bezugs bei Variablen und Feldern

3. Die Namen der Variablen und Felder mit begrenzter Wechselwirkung enden auf X.
Gemeint sind die Groessen, die nur mit einigen Unterprogrammen in Verbindung stehen, jedoch nicht in oder unterhalb von STEBA benoetigt werden.
4. Zentrale Felder und Variablen enden auf Y.
Als zentrale Felder oder zentrale Variablen werden Groessen bezeichnet, die in STEBA, d.h. durchgehend im gesamten Rechenlauf, zur Verfuegung gestellt, veraendert oder erzeugt werden.

A 3.2 DIMENSIONIERUNGSPARAMETER

Die Dimensionierung der Felder in den COMMON-Blocken erfolgt durch Dimensionierungsparameter, deren Name entsprechend den Regeln in Kapitel A 3-1 mit L beginnt. Die Werte der Dimensionierungsparameter werden in PARAMETER-Anweisungen zugewiesen. Die PARAMETER-Anweisungen sind in einem File zusammengefasst, das mit dem logischen Namen 'DIMPARAMETER' angesprochen wird. Mithilfe einer INCLUDE-Anweisung werden die Zeilen in dem File bei der Uebersetzung der Unterprogramme an der entsprechenden Stelle eingefuegt.

Diese Organisation erleichtert eventuelle Aenderungen an den Dimensionierungsparametern, die nunmehr lediglich in dem PARAMETER-File vorgenommen werden muessen. Derzeit sind noch nicht alle COMMON-Blocke mit den Parametern dimensioniert. Die bereits verwendeten Dimensionierungsparameter sind mit ihren aktuellen Werten in der folgenden Tab. A 3-2 zusammengestellt. Die angegebenen Zahlen sind als Obergrenzen zu verstehen und nicht mit der aktuellen Anzahl der benoetigten Belegungen in einem Feld zu verwechseln. Die Bedeutung der Buchstaben in der ersten Spalte der Tab. A 3-2 ergibt sich aus der Tab. A 3-1. Die letzten beiden Dimensionierungsparameter beziehen sich lediglich auf lokale Speicherbereiche in dem Unterprogramm ADATIN (siehe Kap. A 2.4.6).

Speicher- bereich	Name	Inhalt	Wert
N	LNK	Maximalzahl betrachteter Nuklide	80
	LCNK	Maximalzahl der CHARACTER-Angaben fuer jedes Nuklid	1
	LIN	Maximalzahl der INTEGER-Angaben fuer jedes Nuklid	4
	LRN	Maximalzahl der REAL-Angaben fuer jedes Nuklid	5
E	LE	Maximalzahl betrachteter Elemente	30
	LE	Maximalzahl der REAL-Angaben fuer jedes Element	25
	LE	Maximalzahl der CHARACTER-Angaben fuer jedes Element	1
S	LS	Maximalzahl der Strukturvektoren, d.h. der Zeilen unter dem Stichwort **STRUKTUR im Job-Input-File	40
	LIS	Maximalzahl der Komponenten eines Strukturvektors	9
	LB	Maximalzahl betrachteter Barrieren	70
	LCB	Maximalzahl der CHARACTER-Angaben fuer jede Barriere	2
	LCB	Maximalzahl der CHARACTER-Angaben fuer jede Barriere bei der Zwischenspeicherung	2

	LIB1Y	Maximalzahl der allgemeinen INTEGER-Angaben fuer jede Barriere	4
	LIB2Y	Maximalzahl barrierenspezifischer INTEGER-Eingangsdaten	24
B	LRB1Y	Maximalzahl der allgemeinen Barrierendaten fuer jede Barriere	5
	LRB2Y	Maximalzahl barrierenspezifischer REAL-Eingangsdaten	20
	LRB3Y	Maximalzahl der barrierenspezifischen, zeitabhaengigen Daten	25
	LRB4Y	Maximalzahl der nuklidspezifischen Datensaeetze pro Barriere	3
<hr/>			
U	LRU2Y	Maximalzahl anderer Uebergangstroeme	15
	LRU3Y	Maximalzahl anderer Uebergabedaten	15
<hr/>			
A	LRA2Y	Maximalzahl allgemeiner Abfalldaten	5
<hr/>			
	LFY	Maximalzahl der betrachteten Freisetzungintervalle	5
F	LRF1Y	Maximalzahl der REAL-Groessen pro Freisetzungintervall	2
<hr/>			
D	LDY	Zur Zeit ohne Bedeutung	5
	LRDY	Zur Zeit ohne Bedeutung	6
	LIDY	Zur Zeit ohne Bedeutung	1
<hr/>			
Speicherbereich fuer Abfallmixturen (lokal in ADATIN)	LAM	Maximalzahl der Abfallmixturen in dem Job-Input-File	20
	LCIA	Maximalzahl der Einzelabfaelle in einer Abfallmixture	50

 Tab. A 3-1: Dimensionierungsparameter mit aktueller Belegung

A 3.3 BESCHREIBUNG DER COMMON - BLOECKE

In seinem derzeitigen Entwicklungsstand werden in dem Programm EMOS2 27 COMMON-Blocke verwendet. Sie lassen sich im wesentlichen in 5 Gruppen zusammenfassen:

- 6 COMMON-Blocke mit Steuergroessen
- 6 COMMON-Blocke mit lokalen Eingangsdatenfeldern
- 10 COMMON-Blocke mit zentralen Eingangsdatenfeldern
- 2 COMMON-Blocke auf Barrierenmodellebene
- 3 COMMON-Blocke mit Ergebnissen

In den COMMON-Blocken mit zentralen Eingangsdatenfeldern werden auch aktuelle Werte zeitabhaengiger Groessen und Teilergebnisse der Rechnungen abgelegt. Die Reihenfolge der COMMON-Blocke ist an der Gruppenzugehoerigkeit und am Programmablauf orientiert.

Fuer die Namen der COMMON-Blocke gelten in Anlehnung an die Konventionen ueber die Namensgebung der Variablen folgende Vereinbarungen:

1. COMMON-Block-Namen bestehen aus 4 Buchstaben.
2. COMMON-Block-Namen beginnen mit W (WORKSPACE).
3. Der zweite Buchstabe des COMMON-Block-Namens ist
 - C fuer COMMON-Blocke, die CHARACTER-Daten enthalten.
 - Z fuer COMMON-Blocke, die REAL- oder INTEGER-Daten enthalten.
4. Der dritte Buchstabe des COMMON-Block-Namens kennzeichnet den inhaltlichen Zusammenhang der Felder in dem COMMON-Block (vgl. Tab. A 3-1).
5. Namen der COMMON-Blocke, die in einigen Unterprogrammen, jedoch nicht in oder unterhalb von STEBA benoetigt werden, enden auf X.
6. Namen der COMMON-Blocke, die ueberwiegend zentrale Felder oder Variablen enthalten, enden auf Y.

Diese Regeln sind derzeit noch nicht fuer alle COMMON-Block-Namen realisiert.

Beschreibung des COMMON-Blocks: / /
=====

Inhalt : Hilfsgrößen zur Handhabung der Datenbibliothek und Feststellung des Rechenzeitverbrauches

Laenge : 440 Bytes

Dokumentationsstand : 15.03.1986

Uebergeordnetes Programm : EMOS

Programme mit schreiben-
dem Zugriff : EMOS2,HR,PERMLIS

Variablenliste : LFILE(100),IST(6),ZEIT,ZEITA

Bedeutung der Variablen :

LFILE : Zahl der benötigten Blöcke zur Abspeicherung der Daten in der Datenbibliothek

ZEIT : Aktuelle Rechenzeit

ZEITA : Zeit zu Beginn eines Moduls

IST(1) : Nummer des Datenfiles mit nuklidspezifischen Daten

IST(2) : Nummer des Datenfiles mit elementspezifischen Daten

IST(3) : Nummer des Datenfiles mit abfallspezifischen Daten

IST(4) : Nummer des Datenfiles mit barrierenspezifischen Daten

IST(5) : Zur Zeit ohne Bedeutung

IST(6) : Zur Zeit ohne Bedeutung

Beschreibung des COMMON-Blocks: /RC/
=====

Inhalt : CHARACTER Variablen zur Ausdrucksteuerung
Laenge : 370 Bytes
Dokumentationsstand : 15.03.1986
Uebergeordnetes Programm : STEUDA
Programme mit schreiben-
dem Zugriff :
Variablenliste : CNAU(10), CNTHR(5), CNNPR(5), CNUR(5), CNAMR(5),
CNPL(10), CBAU(14), CNAZEF, CBPL(14)
Bedeutung der Variablen :
CNAU : Namen ausgewählter Nuklide fuer Ergebnis-
ausdruck
CNTHR : Nuklidnamen zur Bildung einer Massensumme
fuer die Thorium-Reihe
CNNPR : Nuklidnamen zur Bildung einer Massensumme
fuer die Neptunium-Reihe
CNUR : Nuklidnamen zur Bildung einer Massensumme
fuer die Uran-Reihe
CNAMR : Nuklidnamen zur Bildung einer Massensumme
fuer die Americium-Reihe
CNPL : Zur Zeit ohne Bedeutung
CBAU : Namen ausgewählter Barrieren fuer Ergebnis-
ausdruck
CNAZEW : Name der Barriere, fuer die Freisetzungsmen-
gen ausgedruckt werden sollen
CBPL : Zur Zeit ohne Bedeutung

Beschreibung des COMMON - Blocks /RZ/
=====

Inhalt : Variablen zur Ausdrucksteuerung

Laenge : 600 Bytes

Dokumentationsstand : 15.03.1986

Uebergeordnetes Programm: REPOS

Programme mit schreibendem Zugriff : STEBA, STEUDA, DALEKO

Variablenliste : IAU(10), JAU(10), IIN(10), IDR(10), ZDR(5), NDR(5), IOU(10), ITHR(5), INPR(5), IUR(5), IAMR(5), IME(10), INAZEF, ZEFT(5), IPL(10), IBPL(14), INPL(10), JPL(10)

Bedeutung der Variablen :

IAU : Steuerparameter zur Auswahl von Barrieren und Nukliden fuer Ergebnisausdruck

JAU : Atomgewichte der ausgewaehlten Nuklide

IIN : Steuerparameter zum Drucken der Datensaeetze von der Bibliothek oder dem Job-Input-File

IDR : Steuerparameter fuer den zeitabhaengigen Ausdruck

ZDR : Obere Grenze der Zeitbereiche, in denen der zeitabhaengige Ausdruck unterschiedlich oft erfolgt

NDR : Haeufigkeiten des zeitabhaengigen Ausdrucks in den Zeitbereichen

IOU : Steuerparameter fuer den Ausdruck kumulierter Freisetzungsmengen am Ende der Rechnung

ITHR : Atomgewichte der Nuklide zur Bildung einer Massensumme fuer die Th-Reihe

INPR : Atomgewichte der Nuklide zur Bildung einer Massensumme fuer die Np-Reihe

IUR : Atomgewichte der Nuklide zur Bildung einer Massensumme fuer die U-Reihe

IAMR : Atomgewichte der Nuklide zur Bildung einer Massensumme fuer die Am-Reihe

IME : Steuerparameter zum Ausdrucken der Freisetzungstabellen am Ende der zeitdiskreten Rechnung

INAZEF : Nummer der Barriere, fuer die Freisetzungsmengen bzw. -raten ausgedruckt werden sollen

ZEFT : Obere Grenzen der Zeitbereiche, fuer die Freisetzungsmengen bzw. -raten ausgedruckt werden sollen

IPL, IBPL, INPL, JLP : Zur Zeit ohne Bedeutung

Beschreibung des COMMON-Blocks: /YC/
=====

Inhalt : CHARACTER-Variablen fuer die Zeitschritt-
steuerung

Laenge : 30 Bytes

Dokumentationsstand : 15.03.1986

Uebergeordnetes Programm: REPOS

Programme mit schrei-
bendem Zugriff : STEBA

Variablenliste : CBANA, CBANAA, CNEFF

Bedeutung der Variablen :

CBANA : Name der aktuellen Barriere

CBANAA : Name der Barriere, in deren zugehoerigem
Barrierenmodell ein Einzeleffekt aufgerufen
wurde, der Einfluss auf die Zeitschrittsteue-
rung genommen hat

CNEFF : Name des Effektes, der Einfluss auf die Zeit-
schrittsteuerung genommen hat

Beschreibung des COMMON Blocks: /YZ/
=====

Inhalt : Variablen zur Zeitschrittsteuerung

Laenge : 288 Bytes

Dokumentationsstand : 15.03.1986

Uebergeordnetes Programm: REPOS

Programme mit schreibendem Zugriff : DALEKO, STEBA, ZWLAG, MOBZ5, VQUELLE1

Variablenliste : TL, TA, TN, DT, DT1, DTA, DTMA, DTMI, DASZ, NSTEP, EPSI, IW, TNE, DDTU, DDTO, ANTKM, TB, KB, IY (39)

Bedeutung der Variablen :

TL : Zeitpunkt des Laugenzutritts

TA : Beginn des aktuellen Zeitschrittes

TN : Ende des aktuellen Zeitschrittes

DT : Zeitschrittweite

DT1 : Anfangs- und minimale Schrittweite

DTA : Alte Zeitschrittweite

DTMA : Maximale Zeitschrittweite des naechsten Zeitschrittes

DTMI : Minimale Zeitschrittweite des naechsten Zeitschrittes

DASZ : Dauer des Szenarios

NSTEP : Nummer des Zeitschrittes

EPSI : Konstante zur Berechnung des Bereiches der relativen Stromaenderung, der zu keiner Aenderung der Schrittweite fuehrt

IW : Wiederholungsfaktor der Barriere

TNE : Barrierenspezifisch festgelegter naechster Zeitpunkt, der als Intervallgrenze eines Zeitschrittes getroffen werden soll

DDTU : Relative Schrittweitenverkleinerung

DDTO : Relative Schrittweitenvergroesserung

ANTKM : Minimaler Anteil des aktuellen Freisetzungstromes an der kumulierten Freisetzungsmenge, damit die relative Stromaenderung Einfluss auf die Schrittweitensteuerung hat

TB : Dauer der Betriebsphase

KB : Name der aktuellen Barriere

IY : Zur Zeit ohne Bedeutung

Beschreibung des COMMON-Blocks: /WTEXT/
=====

Inhalt : CHARACTER-Hilfsgroessen

Laenge : 90 Bytes

Dokumentationsstand : 15.03.1986

Uebergeordnetes Programm: REPOS

Programme mit schreibendem Zugriff : REPOS

Variablenliste : CTEXTI, CBLANK

Bedeutung der Variablen :

CTEXTI : Text der Kommentarzeile aus dem Job-Input-File

CBLANK : BLANK-String

Beschreibung des COMMON-Blocks: /WCAX/

=====

Inhalt : CHARACTER-Groessen zur Zwischenspeicherung
eines abfallspezifischen Datensatzes von
der Bibliothek

Laenge : 10 Bytes

Dokumentationsstand : 15.03.1986

Uebergeordnetes Programm: DALEKO

Programme mit schrei-
bendem Zugriff : ABDAIN

Variablenliste : CAX

Bedeutung der Variablen :

CAX : Abfallname

Beschreibung des COMMON-Blocks: /WZAX/

=====

Inhalt : Zwischengespeicherter abfallspezifischer Datensatz von der Bibliothek

Laenge : 680 Bytes

Dokumentationsstand : 15.03.1986

Uebergeordnetes Programm: DALEKO

Programme mit schreibendem Zugriff : ABDAIN

Variablenliste : RAB1X(LNY), RAB2X(LRAZY)

Bedeutung der Variablen :

RAB1X : Aktivitaetsinventar des Abfallgebundes

RAB2X(1) : Matrixmasse des Abfallgebundes

RAB2X(2) : Behaeltermasse des Abfallgebundes

RAB2X(3) : Volumen des Abfallgebundes

RAB2X(4) : Hohlraumvolumen des Abfallgebundes

RAB2X(5) : Zwischenlagerzeit des Abfallgebundes

Beschreibung des COMMON-Blocks: /WCBX/

=====

Inhalt : CHARACTER-Variablen zur Zwischenspeicherung
barrierenspezifischer Daten

Laenge : 1400 Bytes

Dokumentationsstand : 15.03.1986

Uebergeordnetes Programm: DALEKO

Programme mit schrei-
bendem Zugriff : BMIN

Variablenliste : CBX(LCBX,LBY)

Bedeutung der Variablen :

CBX(1,KBY) : Name des in der Barriere eingelagerten
Abfalls

CBX(2,KBY) : Name der Barriere, deren Eingangsdaten
zu uebernehmen sind

Beschreibung des COMMON-Blocks: /WZBX/

=====

Inhalt : Zwischengespeicherter, barrierenspezifischer Datensatz

Laenge : 304 Bytes

Dokumentationsstand : 15.03.1986

Uebergeordnetes Programm : DALEKO

Programme mit schreibendem Zugriff : BADAIN, BDATIN

Variablenliste : RB1X(LRB1Y), RB2X(LRB2Y), IB1X(LIB1Y-2), IB2X(LIB2Y)

Bedeutung der Variablen :

IB1X(1) : Wiederholungsfaktor der Barriere

IB1X(2) : Anzahl der Abfaelle in der Mischung

RB1X : Allgemeine REAL-Barrierendaten (vgl. RB1Y in /WZBY/)

IB2X : Barrierenspezifische INTEGER-Eingangsdaten (vgl. IB2Y in /WZBY/)

RB2X : Barrierenspezifische REAL-Eingangsdaten (vgl. RB2Y in /WZBY/)

Beschreibung des COMMON-Blocks: /WCB1X/

=====

Inhalt : Zwischengespeicherte CHARACTER-
Groessen eines barrierenspezifischen
Datensatzes

Laenge : 40 Bytes

Dokumentationsstand : 15.03.1986

Uebergeordnetes Programm: DALEKO

Programme mit schrei-
bendem Zugriff : BADAIN, BDATIN

Variablenliste : CB1X(LCBY+LCBX)

Bedeutung der Variablen :

CB1X(1) : Name der Barriere

CB1X(2) : Name des Barrierenmodells

CB1X(3) : Name des in der Barriere eingelagerten Ab-
falls

CB1X(4) : Name der Barriere, deren Eingangsdaten
zu uebernehmen sind

Beschreibung des COMMON-Blocks: /WCSX/
=====

Inhalt : CHARACTER-Groessen einer Strukturzeile
beim Einlesen von dem Job-Input-File

Laenge : 3200 Bytes

Dokumentationsstand : 15.03.1986

Uebergeordnetes Programm: DALEKO

Programme mit schreibendem Zugriff : SDATIN

Variablenliste : CSX(LISY-1,LSY)

Bedeutung der Variablen :

CSX(1,KSY) : Name der aufnehmenden Barriere

CSX(2,KSY)
: Namen der liefernden Barrieren

CSX(NISY-1,KSY)

Beschreibung des COMMON Blocks: /WCNY/
=====

Inhalt : CHARACTER-Variablen der nuklidspezifischen
Daten

Laenge : 160 Bytes

Dokumentationsstand : 15.03.1986

Uebergeordnetes Programm: REPOS

Programme mit schrei-
bendem Zugriff : NUDAIN

Variablenliste : CNY(LCNY,LNY)

Bedeutung der Variablen :

CNY(1,KNY) : Nuklidnamen

Beschreibung des COMMON-Blocks: /WZNY/

=====

Inhalt : Nuklidspezifische Daten

Laenge : 4504 Bytes

Dokumentationsstand : 15.03.1986

Uebergeordnetes Programm: REPOS

Programme mit schreibendem Zugriff : ELDAIN, NUDAIN

Variablenliste : RNY(LRNY, LNY), INY(LINY, LNY), NNY, NNR(5)

Bedeutung der Variablen :

IN(1,KNY) : Atomgewicht

IN(2,KNY) : Zeilenabstand des Tochternuklides in den Nuklidmatrizen

IN(3,KNY) : Strahlungstyp

IN(4,KNY) : Spaltennummer des zugehoeriges Elementes in den Elementmatrizen

RNY(1,KNY) : Halbwertszeit

RNY(2,KNY) : Zerfallsrate

RNY(3,KNY) : Dosiskonversionsfaktor

RNY(4,KNY) : Waermekonversionsfaktor

RNY(5,KNY) : Massenkonzentrationsfaktor

NNRY(1) : Anzahl beruecksichtigter Spaltprodukte

NNRY(2) : Anzahl beruecksichtigter Nuklide der Th-Reihe

NNRY(3) : Anzahl beruecksichtigter Nuklide der Np-Reihe

NNRY(4) : Anzahl beruecksichtigter Nuklide der U-Reihe

NNRY(5) : Anzahl beruecksichtigter Nuklide der Am-Reihe

Beschreibung des COMMON-Blocks: /WCEY/
=====

Inhalt : CHARACTER-Groessen fuer elementspezifische
Daten

Laenge : 1060 Bytes

Dokumentationsstand : 15.03.1986

Uebergeordnetes Programm: REPOS

Programme mit schreibendem Zugriff : ELDAIN

Variablenliste : CEY(LCEY,LEY),CETEXT(LREY)

Bedeutung der Variablen :

CEY(1,KEY) : Elementname

CETEXT(KREY) : Bezeichnung des Datentyps in der KREY-ten
Zeile der Matrix REY

Beschreibung des COMMON-Blocks: /WZEY/
=====

Inhalt : Elementspezifische Daten
Laenge : 6008 Bytes
Dokumentationsstand : 15.03.1986
Uebergeordnetes Programm: REPOS
Programme mit schreibenden Zugriff : ELDAIN
Variablenliste : REY(LREY,LEY),NEY,NREY
Bedeutung der Variablen :
REY(KREY,KEY) : Elementspezifische Daten aus dem entsprechenden Datenfile
NREY : Aktuelle Zahl von elementspezifischen Daten pro Element
NEY : Aktuelle Zahl von Elementen

Beschreibung des COMMON-Blocks: /WZSY/

=====

Inhalt : Daten der Strukturmatrix

Laenge : 1448 Bytes

Dokumentationsstand : 15.03.1986

Uebergeordnetes Programm: REPOS

Programme mit schreibendem Zugriff : BARINV,KSEBUE,FRBAST,SDATIN

Variablenliste : ISY(LISY,LSY),NSY,NISY

Bedeutung der Variablen :

ISY(1,KSJ) : Anzahl der liefernden Barrieren

ISY(2,KSJ) : Spaltennummer der aufnehmenden Barriere in der Matrix CBY

ISY(3,KSJ) : Spaltennummer der liefernden Barrieren in der Matrix CBY

:

ISY(NISY,KSJ)

NISY : Um 1 erhoehnte maximale Laenge der Strukturvektoren, d.h. der Zeilen unter dem Stichwort **STRUKTUR im Job-Input-File

NSY : Anzahl der Strukturvektoren

Beschreibung des COMMON-Blocks: /G/
=====

Inhalt : Globale Barrierendaten
Laenge : 800 Bytes
Dokumentationsstand : 15.03.1986
Uebergeordnetes Programm: REPOS
Programme mit schreibendem Zugriff : BDATIN
Variablenliste : G(100)
Bedeutung der Variablen :
G : Globale Barrierendaten
(siehe Kap. D 2.5.1)

Beschreibung des COMMON-Blocks: /WCBY/
=====

Inhalt : Barrierenspezifische CHARACTER-Groessen

Laenge : 1400 Bytes

Dokumentationsstand : 15.03.1986

Uebergeordnetes Programm: REPOS

Programme mit schreibendem Zugriff : BMIN

Variablenliste : CBY(LCBY,LBY)

Bedeutung der Variablen :

CBY(1,KBY) : Name der Barriere

CBY(2,KBY) : Name des Barrierenmodells

Beschreibung des COMMON-blocks :/WZBY/
=====

Inhalt : Barrierenmatrizen
Laenge : 304 Bytes
Dokumentationsstand : 15.03.1986
Uebergeordnetes Programm: REPOS
Programme mit schreibendem Zugriff : BADAIN, ZEITEN, BDATIN, BMIN, KSEBUE, BMOD
Variablenliste : RB1Y(LRB1Y,LBY),RB2Y(LRB2Y,LBY),
RB3Y(LRB3Y,LBY),RB4Y(LNY,LRB4Y,LBY),
IB1Y(LIB1Y,LBY),IB2Y(LIB2Y,LBY),NBY
NRB2Y,NRB3Y

Bedeutung der Variablen :

Allgemeine REAL-Barrierendaten

RB1Y(1,KBY) : Einlagerungszeitpunkt / Zeitpunkt des Abwerfens
RB1Y(2,KBY) : Beginn des Laugenzuflusses
RB1Y(3,KBY) : Ende des Laugenzuflusses
RB1Y(4,KBY) : Ende der Mobilisierung bzw. Zeitpunkt des Erreichens der Endporositäet
RB1Y(5,KBY) : Freie REAL-Variable
RB2Y(KRB2Y,KBY) : Barrierenspezifische REAL-Eingangsdaten
RB3Y(KRB3Y,KBY) : Barrierenspezifische zeitabhaengige Variablen
RB4Y(. ,KRB4Y,KBY) : Barrierenspezifische Nukliddatensaetze

Allgemeine INTEGER-Barrierendaten

IB1Y(1,KBY) : Nummer der Spalte in der Strukturmatrix ISY, deren zweite Komponente KBY ist (bzw. 0 fuer Barrieren ohne Eingaenge)
IB1Y(2,KBY) : Nummer der Spalte der uebergeordneten Barriere in der Matrix CBY (bzw. 0 fuer die Wurzelbarriere)
IB1Y(3,KBY) : Wiederholungsfaktor der Barriere
IB1Y(4,KBY) : Anzahl der Abfaelle in der Mixtur (bzw. 1 fuer die Einzelabfaelle)
IB2Y : Barrierenspezifische INTEGER-Eingangsdaten
NBY : Aktuelle Anzahl der Barrieren
NRB2Y : Aktuelle Anzahl der REAL-Eingangsdaten
NRB3Y : Aktuelle Anzahl barrierenspezifischer, zeitabhaengiger Variablen

Beschreibung des COMMON-Blocks: /WZAY/
=====

Inhalt : Abfallmatrizen

Laenge : 47600 Bytes

Dokumentationsstand : 15.03.1986

Uebergeordnetes Programm: REPOS

Programme mit schreibendem Zugriff : ADATIN, BARINV

Variablenliste : RA1Y(LNY,LBY),RA2Y(LRAZY,LBY)

Bedeutung der Variablen :

RA1Y : Aktivitaetsinventar der Barriere

RA2Y(1,KBY) : Matrixmasse) ggf. gemittelt ueber die)

RA2Y(2,KBY) : Behaeltermasse) entsprechende Groessen)

RA2Y(3,KBY) : Gebindevolumen) der Einzelabfaelle, falls)

RA2Y(4,KBY) : Hohlraumvolumen) die Barriere eine Abfall-)

RA2Y(5,KBY) : Zwischenlagerzeit) mixtur enthaelt

Beschreibung des COMMON-Blocks: /WZIY/
=====

Inhalt : Barriereninventar-Matrix
Laenge : 44800 Bytes
Dokumentationsstand : 15.03.1986
Uebergeordnetes Programm: REPOS
Programme mit schreibendem Zugriff : BARINV
Variablenliste : RIY(LNY,LBY)
Bedeutung der Variablen :
RIY(KNY,KBY) : Aktivitaetsinventare aller Barrieren
des Systems zu Beginn der Nachbetriebsphase

Beschreibung des COMMON-Blocks: /N/
=====

Inhalt : CHARACTER-Variablen fuer die Ausgabe von
Uebergangsdaten

Laenge : 300 Bytes

Dokumentationsstand : 15.03.1986

Uebergeordnetes Programm: STEBA

Programme mit schrei-
bendem Zugriff : MOBZ5, VQUELLE1

Variablenliste : CUNAME(30)

Bedeutung der Variablen :

CUNAME : Namen der Uebergangstroeme und der weiteren
Uebergabedaten

Beschreibung des COMMON-Blocks: /E/
=====

Inhalt : Barrierenspezifische Daten

Laenge : 360 Bytes

Dokumentationsstand : 15.03.1986

Uebergeordnetes Programm: BMOD

Programme mit schreibendem Zugriff : BMOD, VQUELLE1

Variablenliste : RED1(LRB2Y), DVP1(LRB3Y)

Bedeutung der Variablen :

RED1 : Barrierenspezifische REAL-Eingangsdaten
(vgl. RB3Y in /WZBY/)

DVP1 : Barrierenspezifische zeitabhaengige Variablen
(vgl. RB3Y in /WZBY/)

Beschreibung des COMMON-Blocks: /WZUY/

=====

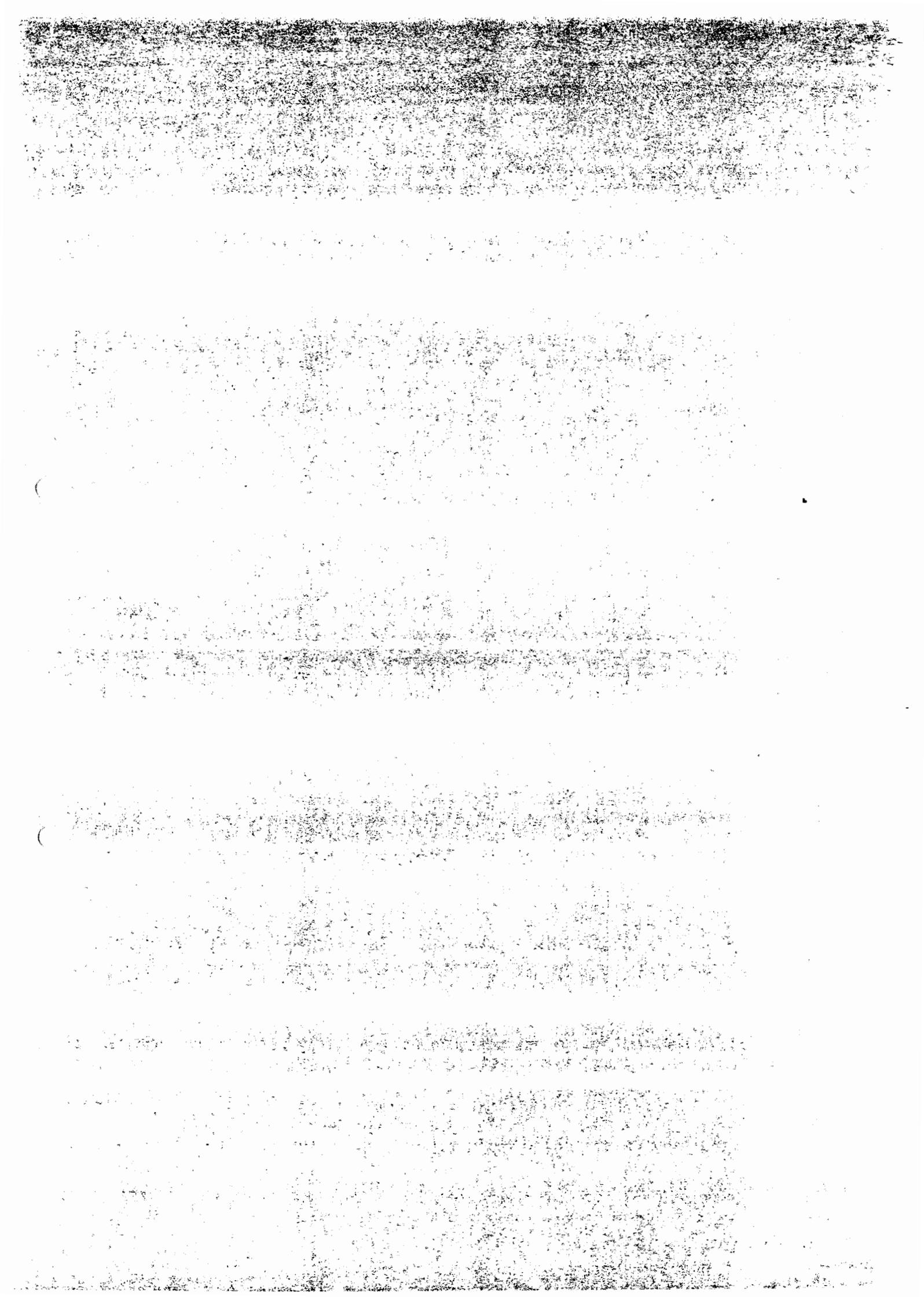
Inhalt : Uebergabematrizen
Laenge : 61608 Bytes
Dokumentationsstand : 15.03.1986
Uebergeordnetes Programm: REPOS
Programme mit schreibendem Zugriff : DALEKO, BMOD
Variablenliste : RU1Y(LNY,LBY), RU2Y(LRU2Y,LBY), RU3Y(LRU3Y,LBY), NRU2Y, NRU3Y
Bedeutung der Variablen :
RU1Y(KNY,KBY) : Aktivitaetsausgangsstroeme
RU2Y(KRU2Y,KBY) : andere Uebergangstroeme
RU3Y(KRU3Y,KBY) : andere Uebergabedaten
NRU2Y : Aktuelle Anzahl anderer Uebergangstroeme
NRU3Y : Aktuelle Anzahl anderer Uebergabedaten

Beschreibung des COMMON-Blocks: /WZKY/
=====

Inhalt : Kumulierte Groessen
Laenge : 53200 Bytes
Dokumentationsstand : 15.03.1986
Uebergeordnetes Programm: REPOS
Programme mit schreibendem Zugriff : DALEKO, STEBA
Variablenliste : RK1Y(LNY,LBY), RK2Y(LRU2Y,LBY)
Bedeutung der Variablen :
RK1Y(KNY,KBY) : Kumulierte Mengen aus der Barriere freigesetzter Aktivitaeten
RK2Y(KRU2Y,KBY) : Kumulierte Mengen aus der Barriere freigesetzter Materialien

Beschreibung des COMMON-Blocks: /WZFY/
=====

Inhalt : Freigesetzte Aktivitaeten und Loesungsstroeme
Laenge : 3284 Bytes
Dokumentationsstand : 15.03.1986
Uebergeordnetes Programm: STEBA
Programme mit schreibendem Zugriff : STEBA
Variablenliste : RF1Y(LFY, LRF1Y), RF2Y(LFY, LNY), NFY, NRF1Y
Bedeutung der Variablen :
RF1Y(KFY, 1) : Zur Zeit ohne Bedeutung
RF1Y(KFY, 2) : Im KFY-ten Zeitintervall ausgepresster Loesungsstrom
RF2Y : Mittlere Freisetzungsrates eines Nuklids in KFY-ten Zeitintervall
NFY : Aktuelle Anzahl von Freisetzungsintervallen



TEIL B: STRUKTUREN UND HANDHABUNG DER DATENBIBLIOTHEK

B 1 DATENBIBLIOTHEK

Die Eingabe groesserer Datenmengen in das Programm EMOS2 erfolgt aus einer Datenbibliothek, in der im wesentlichen vier Arten von Datenfiles enthalten sind:

1. nuklidspezifische Daten (Typ: NUDA)
2. elementspezifische Daten (Typ: ELDA)
3. abfallspezifische Daten (Typ: ABDA)
4. barrierenspezifische Daten (Typ: BADA)

Die Eingabe der barrierenspezifischen Daten kann wahlweise auch ueber das Job-Input-File erfolgen (siehe Kap. D).

B 1.1 VERWALTUNG DER DATENBIBLIOTHEK

Neben den bereits erwaehten Daten enthaelt die Datenbibliothek in der Directory [EMOS.DATEN] zusaetzlich ein INDEX-File, in dem die Filebeschreibungen aller Datenfiles eingetragen sind. Die Filebeschreibung eines Datenfiles beinhaltet u.a. auch dessen Filenamem. Die Hauptaufgabe des INDEX-Files besteht darin, einem bestimmten Datenfilenamem eine Nummer zuzuordnen, nach der dieses Datenfile im Rahmen eines Rechenlaufs eindeutig identifiziert werden kann. Beim Aufruf der Nummer (im folgenden als 'Satznummer' bezeichnet), wird der zugehoerige Satz des INDEX-Files mit der Filebeschreibung gelesen und der Filename des Datenfiles ermittelt. Dieses Datenfile wird dann ueber die UNIT-Nummer 11 (nach einer OPEN-Anweisung) im Programmlauf angesprochen. Der Aufbau des INDEX-Files wird in dem nachfolgenden Kap. B 1.2 erlaeuert.

Die Eintragung von neu erzeugten bzw. geaenderten Filebeschreibungen wird durch die DCL-Prozedur JUNK vorgenommen, die sich in der Directory [EMOS.TOOLS] befindet. Die Handhabung der Prozedur JUNK wird in Kap. B 1.4 beschrieben.

Die Anwendung der Prozedur JUNK setzt voraus, dass entsprechende Datenfiles zuvor erzeugt wurden. Eine detaillierte Eingabebeschreibung zu den Datenfiles ist dem Kap. B 1.3 zu entnehmen.

B 1.2 AUFBAU DES INDEX-FILES

Das INDEX-File besteht aus 99 Saetzen (von 1 bis 99). Jeder Satz beinhaltet eine Satznummer, sowie eine Filebeschreibung fuer den Fall, dass die Satznummer mit einem Datenfile belegt ist; andernfalls enthalten die Positionen Leer-Werte. Ein Beispiel fuer den Aufbau eines solchen Satzes zeigt Tab. B 1-1

Bytes	Inhalt	Beispiel
2	Satznummer	48
15	Filename	ELDAKON1.DAT
80	Beschreibung des Inhaltes	ELDAKON1: Elementspezifische Daten fuer KONRAD
23	Datum und Uhrzeit der letzten Aenderung	17-DEC-1985 15:42:31.88
4	Filegroesse (Blocks)	0008
4	Filetype	ELDA
4	Name des Bearbeiters	OTTO

Tab. B 1-1: Aufbau einer Filebeschreibung im INDEX-File

B 1.3 AUFBAU DER DATENFILES IN DER DATENBIBLIOTHEK

Die Datenbibliothek befindet sich, wie auch das INDEX-File in der Directory [EMOS.DATEN]. Veraenderungen des Inhaltes einzelner Datenfiles duerfen grundsatzlich nur unter Beruecksichtigung des jeweils vorgeschriebenen Eingabeformates vorgenommen werden. Aus diesem Grunde wird in den nachfolgenden Unterkapiteln eine detaillierte Eingabebeschreibung vorgelegt. Darin enthalten sind:

- Variablenname bzw. Feldelement, dem ein Wert bei einem EMOS-Rechenlauf zugewiesen werden soll
- Format der Dateneingabe
- textliche Erlaeuterung der Variablen

Die Abarbeitung erfolgt zeilenweise.

B 1.3.1 NUKLIDSPEZIFISCHE DATEN

Die nuklidspezifischen Daten sind ausschliesslich solche Daten, die fuer ein Nuklid charakteristisch sind. In der Datei sind alle zu berueck-sichtigenden Nuklide zu Nuklidgruppen (Spalt- und Aktivierungsprodukte, Th-, Np-, U- und Am- Zerfallsreihe) zusammengefasst. Innerhalb der Gruppe der Spalt- und Aktivierungsprodukte sind die Nuklide aufsteigend nach Massenzahlen sortiert. Innerhalb der Zerfallsreihen ist die Reihen-folge durch die Zerfallskette bedingt. Im Einzelnen werden folgende An-gaben gemacht:

1. Zeile:

Kommentarzeile zur Charakterisierung der nuklidspezifischen Datei

KOMMENTAR[A80]

2. Zeile:

Aufteilung der Nuklide auf Spaltprodukte/Aktivierungsprodukte, bzw. die Zerfallsreihen:

(NNRY(K), K=1,...,5)[515]

NNRY(1) : Anzahl der Spaltprodukte

NNRY(2) : Anzahl der Nuklide in der Th-Reihe

NNRY(3) : Anzahl der Nuklide in der Np-Reihe

NNRY(4) : Anzahl der Nuklide in der U -Reihe

NNRY(5) : Anzahl der Nuklide in der Am-Reihe

Gesamtzahl der beruecksichtigten Nuklide:

NNY = NNRY(1) + ... + NNRY(5) (= LNY (siehe Kap. A 3.2)

3. Zeile bis (NNY+2)-te Zeile:

Angaben zu dem KNY-ten Nuklid:

CNY(1,KNY), (INY(K,KNY), K=1,...,3), (RNY(K,KNY), K=1,...,3) ...
.....[A10,315,3E12.5]

CNY(1,KNY) : Name des Nuklids

INY(1,KNY) : Atomgewicht [g/mol]

INY(2,KNY) : Angaben zu den Zerfallsreihen

=0 : Tochter des Nuklids existiert nicht oder wird nicht betrachtet

>0 : Zeilenabstand des Nuklids zu seiner Tochter

INY(3,KNY) : Zerfallstyp (alpha, beta, gamma)
=0 : beta und/oder gamma
=1 : alpha

RNY(1,KNY) : Halbwertszeit des Nuklids [a]

RNY(2,KNY) : Dosiskonversionsfaktor [(Sv/a)/(Bq/l)]

RNY(3,KNY) : Waermekonversionsfaktor [W/Bq]

B 1.3.2 ELEMENTSPEZIFISCHE DATEN

Das File mit den elementspezifischen Daten enthaelt Angaben, die fuer ein Element charakteristisch sind. Das Datenfile hat folgenden Inhalt:

1. Zeile:

Kommentarzeile zur Charakterisierung der elementspezifischen Datei

KOMMENTAR[A80]

2. Zeile:

Anzahl der Elemente und Anzahl der elementspezifischen Daten je Element:

NEY, NREY[2I5]

NEY : Anzahl der Elemente (\leq LEY, siehe Kap. A 3.2)

NREY : Anzahl der Daten (\leq LREY, siehe Kap. A 3.2)

3. Zeile und folgende:

Textliche Charakterisierung der elementspezifischen Daten:

CETEXT[A40]

CETEXT : textliche Erklarung der elementspezifischen Angabe

Anzahl der Zeilen = NREY

4. Angaben zu jedem Element:

Die Anzahl der nachfolgenden Zeilen richtet sich nach der Anzahl der Elemente

CEY(1,KEY), (REY(K,KEY), K=1,...,NREY) ...[A10,5E12/10X,5E12 usw.]

CEY(1,KEY) : Name des Elements

Fuer K= 1, NREY:

REY(K,KEY) : elementspezifische Daten des Elements CEY(1,KEY)

B 1.3.3 ABFALLSPEZIFISCHE DATEN

Das File mit den abfallspezifischen Daten beinhaltet in der Hauptsache Angaben zum Aktivitaetsinventar eines einzelnen Abfallgebundes. Das Datenfile hat folgende Struktur:

1. Zeile:

Kommentarzeile zur Charakterisierung der abfallspezifischen Datei
KOMMENTAR[A80]

2. Zeile und folgende:

Fuer jeden Gebindetyp werden folgende Eingaben gemacht:

CAX, (RAB1X(K), K=1,...,NNY)[A10,5E12/10X,5E12 ...usw.]

CAX : Name des Abfallgebundes

Fuer K=1,...,NNY:

RA1X(K) : Aktivitaetsinventar zum Zeitpunkt der Herstellung
in Bq/Gebinde

Allgemeine Abfalldaten:

..... (RAB2X(K), K=1,...,5)[10X,5E12]

RAB2X(1) : Matrixmasse [kg]

RAB2X(2) : Behaeltermasse [kg]

RAB2X(3) : Gebindevolumen [m**3]

RAB2X(4) : Hohlraumvolumen [m**3]

RAB2X(5) : Zwischenlagerzeit [a]

B 1.3.4 BARRIERENSPEZIFISCHE DATEN

Das File mit den barrierenspezifischen Daten enthaelt charakteristische Daten fuer jede Barriere.

Sich wiederholende Angaben bezueglich der barrierenspezifischen INTEGER- und REAL-Eingangsdaten (IB2X und RB2X) koennen dabei durch einen Verweis auf andere Barrieren von dort uebernommen werden. Die Reihenfolge der Barrieren richtet sich nach der Reihenfolge der Abarbeitungsebenen in der Strukturmatrix, d.h. Abfallgebinde vor Einlagerungsoerter, vor Einlagerungsstrecken, usw.

1. Zeile:

Kommentarzeile zur Charakterisierung der barrierenspezifischen Datei
KOMMENTAR[A8Ø]

2. Zeile:

Allgemeine Angaben zur Barriere:

CB1X(1), CB1X(2), CB1X(3), CB1X(4), IB1X(1), IB1X(2)[4A1Ø,2I5]

CB1X(1): Name der Barriere

CB1X(2): Name des Barrierenmodells fuer die Barriere

CB1X(3): Name des Abfalls oder der Abfallmixture, falls die Barriere ein Abfallgebinde ist

CB1X(4): Name einer anderen Barriere, von der die Eingangsvektoren IB2X und RB2X uebernommen werden (falls erwuenscht)

IB1X(1): Relative Haeufigkeit dieser Barriere in der naechst-aeusseren Barriere (Wiederholungsfaktor)

IB1X(2): Anzahl der Einzelabfallgebinde in dem Gebinde CB1X(3)

=1: Unter CB1X(3) wird der Name eines Einzelabfallgebundes aus den abfallspezifischen Daten angegeben

>1: Unter CB1X(3) wird der Name eines gemischten Abfallgebundes eingegeben. Die Mischanordnung ist im Job-Input-File festgelegt, Kap. D 2.6

3. Zeile:

Forts.: Allgemeine Angaben zur Barriere:

(RB1X(K), K=1,....,5)[1ØX,5E12.5]

RB1X(1) : Zeitdauer zwischen dem Beginn der Betriebsphase und dem Einlagerungszeitpunkt [a]

RB1X(2) : Rechenergebnis: Beginn des Laugenzuflusses [a]

RB1X(3) : Rechenergebnis: Ende des Laugenzuflusses [a]

RB1X(4) : Rechenergebnis: Ende der Mobilisierung oder Zeitpunkt des Erreichens der Endporositäet

RB1X(5) : nicht benutzt

RB1X(2), RB1X(3), RB1X(4) muessen bei der Eingabe den Wert Null erhalten !

Falls CB1X(4) nicht " ", werden die weiteren Eingangsdaten vom Gebinde CB1X(4) uebernommen. Das Gebinde CB1X(4) darf an beliebiger Stelle in der Liste der barrierenspezifischen Daten stehen.

4. und 5. Zeile:

Angaben zu dem INTEGER-Eingangsdatenvektor IB2X:

(IB2X(K), K=1 ,...,12)[10X,12I5]
(IB2X(K), K=13,....,24)[10X,12I5]

IB2X(1) : Anzahl der REAL-Eingangsdaten (6. Zeile und folgende)

IB2X(2) : Anzahl der nuklidspezifischen Datensaeetze im Feld RB4Y in /WZBY/

IB2X(3) : NRB3Y in /WZBY/, d.h. Anzahl der barrierenspezifischen, zeitabhaengigen Variablen

IB2X(4) : Anzahl der anderen Uebergangstroeme (Feld RU2Y in /WZUY/)

IB2X(5) : Anzahl der weiteren Uebergabedaten (Feld RU3Y in /WZUY/)

IB2X(6) : Anzahl der zu druckenden Positionen der anderen Uebergangstroeme (<= IB2X(4))

IB2X(7) : Anzahl der zu druckenden Positionen der weiteren Uebergabedaten (<= IB2X(5))

IB2X(8) : - z. Zt. ohne Bedeutung -

IB2X(9) : - z. Zt. ohne Bedeutung -

IB2X(10): - z. Zt. ohne Bedeutung -

IB2X(11): - z. Zt. ohne Bedeutung -

IB2X(12): - z. Zt. ohne Bedeutung -

IB2X(13): - z. Zt. ohne Bedeutung -

IB2X(14): Nummer der Zeile der Matrix REY, die die Loeslichkeitsgrenzen enthaelt

IB2X(15): Nummer der Zeile der Matrix REY, die die KD-Werte enthaelt

IB2X(16): Nummer der Zeile der Matrix REY, die die Mobilisierungsdauern enthaelt

IB2X(17): Schalter:

- = 1: wenn es Nuklide in einer Zerfallsreihe gibt, deren zugehoerige Elemente unterschiedliche Mobilisierungsdauern haben
- = 0: sonst

IB2X(18) bis IB2X(24) : nicht benutzt

6. Zeile und folgende:

Angaben zum REAL-Eingangsdatenvektor RB2X:

(RB2X(K), K=1,...,IB2X(1))[10X,5E12.5/10X,5E12.5 ...usw.]

RB2X(1), ... ,RB2X(1B2X(1)) :

Angaben sind barrierenmodell-spezifisch und werden im Barrierenmodell erlaeutert

Alle Angaben wiederholen sich analog fuer jede Barriere mit Beginn der 2. Zeile.

Diese Eingabebeschreibung ist identisch mit der Eingabebeschreibung zu den barrierenspezifischen Daten im Job-Input-File !

B 1.4 HANDHABUNG DER DATENBIBLIOTHEK

Das Eintragen von neuen bzw. geaenderten Datenfiles in das INDEX-File erfolgt ueber die DCL-Prozedur JUNK.COM in der Directory [EMOS.TOOLS]. Die innerhalb dieser Prozedur zur Verfuegung stehenden Funktionen werden nach dem Aufruf der Prozedur angezeigt (siehe Tab. B 1-2). Alle notwendigen Eingaben werden anschliessend interaktiv abgefragt.

Dateiname der Prozedur: [EMOS.TOOLS]JUNK.COM

Aufruf der Prozedur: @[EMOS.TOOLS]JUNK.COM

Parameter im Aufruf: keine

benoetigte Dateien: Datenfiles und INDEX-File in Directory
[EMOS.DATEN]

F U N K T I O N	E I N G A B E
Aendern einer Filebeschreibung	AEN
Loeschen einer Filebeschreibung	LOE
Erzeugen einer neuen Filebeschreibung	ERZ
Listen der Filebeschreibungen aller belegten Satznummern auf Bildschirm	LIS
Suchen nach Eigenschaften (Suchmenue siehe Tab. B 1-3)	SUC
Drucken der Filebeschreibungen aller belegten Satznummern	DRU
Beenden der Prozedur	END

Tab. B 1-2: Hauptmenue in der Prozedur JUNK

Bei den Funktionen AEN, LOE, ERZ wird nach der Satznummer abgefragt und fuer den Fall, dass die Satznummer gueltig ist (1 - 99), ein Hilfsfile INDEX1.DAT eroeffnet, auf das nicht nur die geaenderten, geloeschten oder erzeugten Filebeschreibungen eingetragen werden, sondern auch alle nicht veraenderten Filebeschreibungen. Ist die jeweilige Funktion abgearbeitet, wird das File INDEX.DAT geloescht und das File INDEX1.DAT zu INDEX.DAT umbenannt.

Die Funktion AEN unterscheidet sich von der Funktion ERZ nur dadurch, dass nur belegten Satznummern (Filename # "LEER") ein neuer Filename, Inhalt, Typ und Bearbeiternamen zugewiesen werden kann.

Bei beiden Funktionen werden ausserdem jeder neuen Filebeschreibung Zusatzinformationen mit dem aktuellen Datum, Zeit, sowie mit der Groesse des zu dem geaenderten bzw. erzeugten Filenamens gehoerenden Datenfiles (in Blocks) automatisch angefuegt. Das Anfüegen der Blockgroesse setzt

voraus, dass sich das beschriebene File auch tatsaechlich in der Directory [EMOS.DATEN] befindet!

Die Funktion LOE belegt den Filenamen, Inhalt, Typ und Bearbeiternamen, sowie Datum, Zeit und Anzahl der Blocks mit 'Leer'-Werten.

Die Funktion SUC ruft ein Suchmenue auf, in dem die in Tab. B 1-3 erlaeuterten Suchbefehle zur Verfuegung stehen.

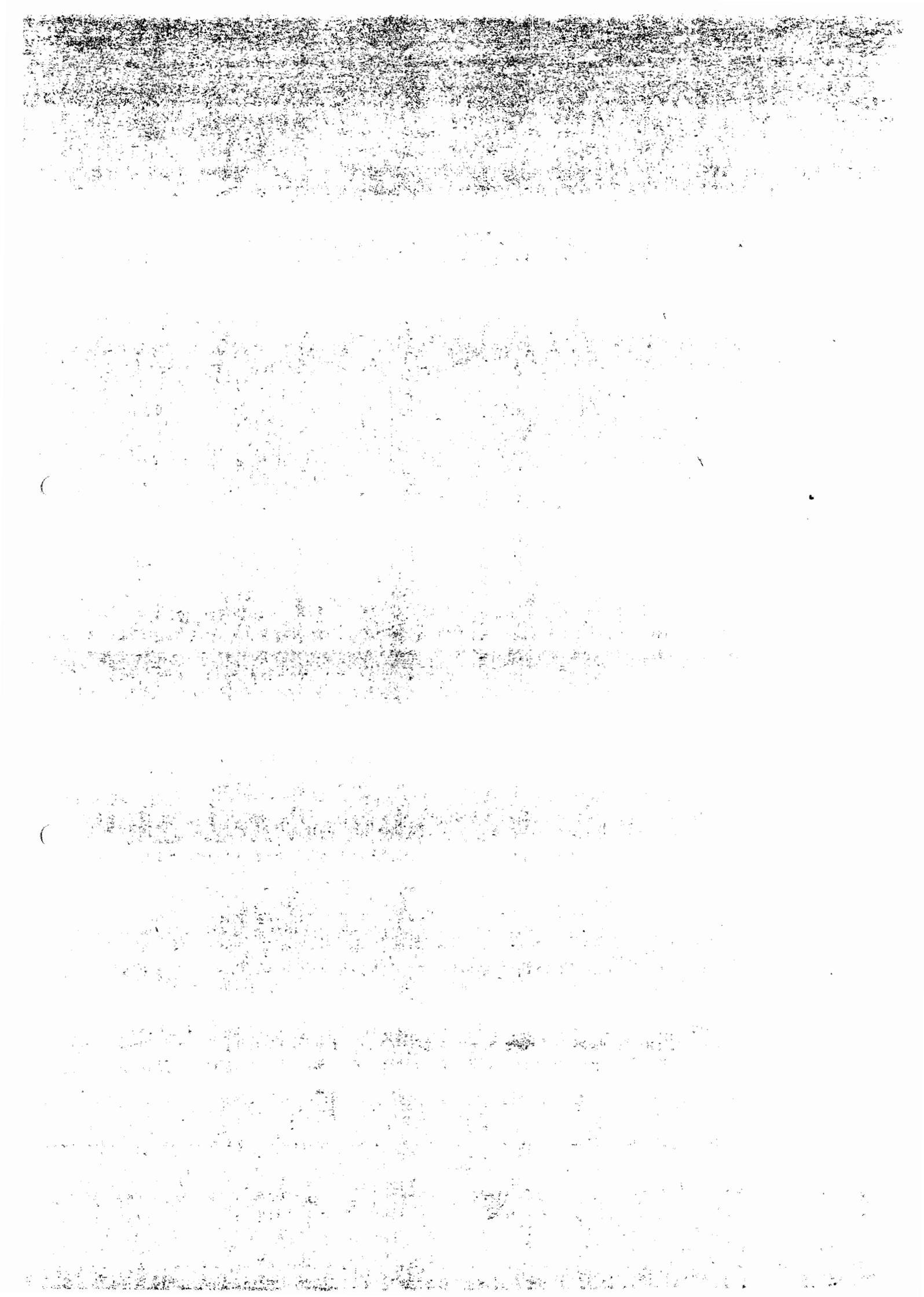
F U N K T I O N	E I N G A B E
Suchen nach Filenamen	FI
Suchen nach Satznummer	SA
Suchen nach Typ	TY
Suchen nach Name des Bearbeiters	NB
Beenden des Suchmenues (zurueck: Hauptmenue)	END

Tab. B 1-3: Suchmenue in der Prozedur JUNK

Die Funktionen FI, SA, TY und NB verlangen nach der Eingabe einer Eigenschaft. Wenn ein Satz mit dieser Eigenschaft vorhanden ist, wird er auf den Bildschirm ausgegeben, andernfalls erfolgt die Meldung: "File nicht vorhanden".

Die Funktion END veranlasst einen Ruecksprung in das Hauptmenue.

Die Arbeitsweise der Prozedur JUNK kann durch das folgende Ablaufdiagramm grob beschrieben werden:



TEIL C: HANDHABUNG DER PROGRAMME

C 1 UEBERSICHT UEBER DIE PROGRAMMDATEIEN

Das Programm EMOS ist in Form von Quelltexten, Daten, Object-Codes und Hilfsprozeduren in einer Vielzahl von Dateien abgelegt. Eine Uebersicht ueber diese Dateien soll in diesem Kapitel gegeben werden. Dazu werden alle aktuellen EMOS2-Dateien aufgefuehrt und deren Inhalt kurz charakterisiert. Die thematische Gliederung laesst sich dabei unmittelbar aus den Directory-Namen ablesen. Eine gewisse Hierarchie wurde durch die Verwendung von Subdirectorys bewusst angestrebt. So sind beispielsweise saemtliche Quelltexte unter der Subdirectory QUELLEN vereint, wobei sich diese wiederum in Subdirectories gliedert.

Der Aufruf einer Datei erfolgt stets ueber deren vollstaendige Filespezifikation. Eine vollstaendige Filespezifikation beinhaltet:

- Name der Directory, ggf. mit Subdirectory (z.B. [EMOS.TOOLS])
- Filename (z.B. RUNEMOS2)
- Erweiterung (z.B. COM)
- Versionsnummer (z.B. 2)

Die Reihenfolge der Eingabe ist wie folgt festgelegt:

[Directory]Filename.Erweiterung;Version

Die Versionsnummer kann entfallen wenn default-maessig auf die letzte Version der Datei zugegriffen werden soll.

Die sogenannte 'Erweiterung' jeder Datei gibt Aufschluss ueber den Dateityp. Es wurden ausschliesslich die nachfolgend aufgelisteten Erweiterungen verwendet:

DIR	: Directory
FOR	: FORTRAN-Quelltext
OBJ	: uebersetztes Programm
OLB	: Objekt-Bibliothek
EXE	: uebersetztes und gebundenes Programm
LOG	: Protokoll eines Rechenlaufs
COM	: Prozedur
DAT	: Daten
DOC	: Dokumentation

Directory-Struktur:

Directory: [EMOS]

BATCH.DIR	Dir.: Dateien fuer Batch-Rechenlaeufe
DATEN.DIR	Dir.: Datenbibliothek
DOKUMENTE.DIR	Dir.: Text-Dateien mit EMOS2-Dokumentation
QUELLEN.DIR	Dir.: EMOS2-Quellprogramme
OBJEKTE.DIR	Dir.: Programmbibliothek mit EMOS2-Object-Codes

JOBS.DIR Dir.: Job-Input-Files
TOOLS.DIR Dir.: Prozeduren (JUNK, RUNEMOS2, ...)

Directory: [EMOS.BATCH]

EMOS2.EXE Lauffaehiges (compiliertes und gelinktes)
 EMOS2-Programm
EMOS2.OBJ Compiliertes Hauptprogramm EMOS2

Directory: [EMOS.DATEN] (Kurzbeschreibung siehe [EMOS.DATEN]INDEX.DAT)

(noch unbesetzt)

Directory: [EMOS.DOKUMENTE]

EMOS2INP.DOC Eingabe-Beschreibung fuer EMOS2-Modul REPOS
EMOS2SPEI.DOC Arbeitsspeicher-Beschreibung (COMMON-Blocke)

Directory: [EMOS.QUELLEN]

BARRIEREN.DIR Quelltexte fuer Barrierenmodelle
EFFEKTE.DIR " " Effekte
MODULE.DIR " " EMOS2-Hauptprogramm und REPOS
PARAMETER.FOR Dimensionierungsparameter
SUBEMOS2.DIR Quelltexte fuer Hilfs- und Serviceroutinen
SUBIO.DIR Quelltexte von Unterprogrammen zum Einlesen und Drucken
 von Eingangsdaten
SUBREPOS.DIR Quelltexte fuer Unterprogramme des Moduls REPOS

Directory: [EMOS.QUELLEN.BARRIEREN]

BMOD.FOR Aufruf der Barrierenmodelle
MOBZ5.FOR Mobilisierungsmodell fuer KONRAD
VQUELLE1.FOR Barrierenmodell fuer KONRAD

Directory: [EMOS.QUELLEN.EFFEKTE]

ELMAKO.FOR Berechnung von Element- und Nuklidmassen sowie

Element- und Nuklidkonzentrationen

RAZE.FOR Berechnung des radioaktiven Zerfalls

SORPRP2.FOR Berechnung von Sorption und Ausfällung

Directory: [EMOS.QUELLEN.MODULE]

EMOS2.FOR Hauptprogramm zur Modellierung von Endlagerszenarien

REPOS.FOR Unterprogramm zur Modellierung der Wirtsformation

Directory: [EMOS.QUELLEN.SUBEMOS2]

HRANF.FOR Hilfsroutine zur Steuerung des Modulablaufs

OPENFIL.FOR Öffnen und Schliessen von Bibliotheksdateien

PERMLIS.FOR Auflisten der Datensätze des INDEX-Files

NOUTBAR.FOR Hilfsfunktion zur Ausdrucksteuerung

Directory: [EMOS.QUELLEN.SUBIO]

ABDAIN.FOR Einlesen der abfallspezifischen Daten (Einzelgebäude)
aus Datenbibliothek

BADAIN.FOR Einlesen der barrierenspezifischen Daten von Bibliothek

DRUABDA.FOR Ausdrucken der abfallspezifischen Daten

DRUBADA.FOR " " barrierenspezifischen Daten

DRUELDA.FOR " " elementspezifischen Daten

DRUFM.FOR " " Freisetzungsmengen und -raten fuer
fuenf Zeitintervalle

DRUNUDA.FOR " " nuklidspezifischen Daten

ELDAIN.FOR Einlesen der elementspezifischen Daten von Bibliothek

NUDAIN.FOR Einlesen von nuklidspezifischen Daten von Bibliothek

PRTGLB.FOR Ausdrucken der globalen Eingabedaten

SDATIN.FOR Eingabe von Strukturdaten

ZEITEN.FOR Ausdrucken einer Tabelle mit den Zulaufzeiten und Zeit-
punkten des Erreichens der Endporositäten bzw. Ende
der Mobilisierung

Directory: [EMOS.QUELLEN.SUBREPOS]

ADATIN.FOR	Aufbau der Abfallmatrix
BARINV.FOR	Berechnung der Inventare des gesamten Barrierensystems
BDATIN.FOR	Aufbau der Barrierenmatrix
BMIN.FOR	Unspeichern barrierenspezifischer Daten und Eintragen in Barrierenmatrix
DALEKO.FOR	Aufruf von Programmen zum Einlesen modellspezifischer Daten
KSEBUE.FOR	Kodieren der Strukturmatrix, Ergaenzen der Barrierenmatrix, Pruefen der Struktur und Vollstaendigkeit
MAXEL.FOR	Erstellen einer Tabelle mit den max. Elementkonzentrationen
PRBAST.FOR	Pruefen, ob die Barriere in Barrierenstruktur vorhanden
SETDTMA.FOR	Setzen der maximalen Zeitschrittweite
SETLG.FOR	Erstellen einer Tabelle mit den Zeitpunkten des Erreichens der Loeslichkeitsgrenzen
STEBA.FOR	Durchfuehrung der zeitdiskreten Rechnung
STEUDA.FOR	Einlesen und Ausdrucken von Steuerparametern
ZWLAG.FOR	Berechnung des radioaktiven Zerfalls waehrend der Zwischenlagerzeit

Directory: [EMOS.OBJEKTE]

EMOS2.OLB Object-Library mit den uebersetzten Quellprogrammen

Directory: [EMOS.JOBS]

noch unbesetzt

Directory: [EMOS.TOOLS]

Hauptprozeduren:

ADDBIB.COM	Uebersetzen einer Programmeinheit und Eintragen in eine bestehende Object-Library
JUNK.COM	Eintragen eines Files der Datenbibliothek in das INDEX-File
LINKEMOS2.COM	Binden des Programms EMOS2 mit Erstellen und Ausdrucken des MAP-Files

NEWBIB.COM Uebersetzen aller Programmeinheiten und Eintragen in
eine neue Object-Library

RUNEMOS2.COM Vorbereiten eines EMOS2-Programmlaufs

Hilfsprozeduren:

BNEWBIB.COM wird von NEWBIB aufgerufen (Batch-Prozedur)

BRUNEMOS2.COM wird von RUNEMOS2 aufgerufen (Batch-Prozedur)

LOGEMOS2.COM Ausdrucken des Protokolls eines Rechenlaufs
(LOG-File), wird von BRUNEMOS2 aufgerufen

Dateien:

QUELLNAM.DAT Datei mit den Namen aller aktuellen EMOS2-Quell-
programme, wird von NEWBIB aufgerufen

C 2 PROZEDUREN

Prozeduren sind lauffaehige Programme, die in einer fuer die Rechenanlage spezifischen Programmiersprache geschrieben sind; hier: DCL (= DIGITAL-Command-Language). Innerhalb dieses Sprachumfangs stehen dem Programmierer alle Systembefehle, sowie eine grosse Anzahl von lexikalischen Funktionen zur String-Verarbeitung zur Verfuegung.

Die Prozeduren, die fuer den EMOS2-Anwender entwickelt wurden, lassen sich grundsaeztlich in zwei Sparten unterteilen:

- Prozeduren, die der Verwaltung der Datenbibliothek (JUNK siehe Kap. B 1.4) und der Programmdateien dienen (ADDBIB, NEWBIB)
- Prozeduren, die zum Starten eines Batch-Rechenlaufs notwendig sind (LINKEMOS2, RUNEMOS2)

Einige Prozeduren rufen weitere Hilfsprozeduren auf, auf die in der nachfolgenden Beschreibung jedoch nicht naeher eingegangen wird. Die Arbeitsweise der Hilfsprozeduren laesst sich jedoch anschaulich aus den Ablaufplaenen der Hauptprozeduren entnehmen.

Der Aufruf einer Prozedur durch den Anwender ist stets mit der Eingabe von Eingangsparametern verbunden, die hinter dem Filenamem mit aufgefuehrt werden. Fuer den Fall, dass keine Eingangsparameter eingegeben werden, werden diese durch die Prozedur nachfolgend abgefragt.

C 2.1 ERGAENZUNG DER OBJEKT-BIBLIOTHEK (ADDBIB)

ADDBIB.COM ist eine DCL Prozedur zum Uebersetzen eines Programmbausteins mit Eintragen in die EMOS2-Object-Library [EMOS.OBJEKTE]EMOS2.OLB. Gleichzeitig wird ein Listing des Quellprogramms ausgegeben.

Dateiname der Prozedur: [EMOS.TOOLS]ADDBIB.COM

Aufruf der Prozedur: @ [EMOS.TOOLS]ADDBIB.COM P1

Parameter im Aufruf: P1= vollstaendige Filespezifikation des einzutragenden Programmbausteins

C 2.2 ERZEUGUNG EINER OBJEKT-BIBLIOTHEK (NEWBIB)

Durch die Prozedur NEWBIB.COM wird es moeglich, dass mehrere Programm-einheiten (i.a. der gesamte EMOS2-Quelltext) bei einem Aufruf von NEWBIB uebersetzt werden koennen, wobei diese Object-Codes dann in eine beliebige, selbst benannte Object-Library eingetragen werden. Die Filespezifikationen der zu uebersetzenden Programmbausteine muessen in einer Datei abgelegt sein, von der sie durch die Prozedur eingelesen werden koennen.

Die Anwendung von NEWBIB ist immer dann hilfreich, wenn zugleich alle EMOS2-Quellprogramme uebersetzt werden muessen. Dies setzt jedoch voraus, dass sich die Datei mit den Filespezifikationen stets auf einem aktuellen Stand befindet.

Dateiname der Prozedur: [EMOS.TOOLS]NEWBIB.COM

Aufruf der Prozedur: @ [EMOS.TOOLS]NEWBIB.COM P1 P2

Parameter im Aufruf: P1= vollstaendige Filespezifikation der Datei mit den Filenamen der zu uebersetzenden Programmbausteine
P2= vollstaendige Filespezifikation der Object-Library

benoetigte Prozeduren: BNEWBIB.COM (Batch-Prozedur)
benoetigte Dateien: Datei mit den vollstaendigen Filespezifikationen der zu uebersetzenden Programmbausteine (z.B. QUELLNAM.DAT)

Ergebnis: Object-Library in P2

C 2.3 BINDEN DES PROGRAMMS (LINKEMOS2)

LINKEMOS2.COM bindet die Programmteile von EMOS2 und erzeugt ein ausfuehrbares EXE-File, welches anschliessend in der Directory [Benutzername.BATCH] steht. Der Name des EXE-Files richtet sich nach dem Filenamen des Object-Codes, welcher das Hauptprogramm EMOS2 beinhaltet. Das Object-File mit dem uebersetzten Hauptprogramm EMOS2 muss in der Directory [Benutzername.BATCH] vorhanden sein. Dieses File kann neben dem Hauptprogramm beliebig viele Unterprogramme enthalten.

Dateiname der Prozedur: [EMOS.TOOLS]LINKEMOS2.COM

Aufruf der Prozedur: @ [EMOS.TOOLS]LINKEMOS2.COM P1 P2 P3

Parameter im Aufruf: P1= Benutzername (z.B. EMOS)
P2= Filename des Object-Codes mit dem Hauptprogramm (default= EMOS2)
P3= vollstaendige Filespezifikation der Object-Library (default= [EMOS.OBJEKTE]EMOS2.OLB)

Ergebnis: ausfuehrbares EXE-File auf
[Benutzername.BATCH]P2.EXE

C 2.4 DURCHFUEHRUNG VON RECHENLAEUFEN (RUNEMOS2)

RUNEMOS2.COM dient zum Starten eines EMOS2-Laufs im Batch-Betrieb. Die Prozedur ermoglicht eine flexible Eingabe der Namen fuer das Job-Input-File und das Ergebnis-File ueber die Parameterliste des Aufrufs. Dadurch ist es auf einfache Weise moeglich, Rechenlaeufe mit verschiedenen Eingabefile- und beliebigen Ergebnisfilenamen zu starten.

Die Prozedur RUNEMOS2.COM setzt voraus, dass bereits ein lauffaehiges EXE-File auf der Directory [Benutzername.BATCH] existiert. Andernfalls muessen die benoetigten Programmteile mit Hilfe der Prozedur LINKEMOS2 vorab gebunden werden.

Das Protokoll-File (LOG-File) steht nach dem Rechenlauf in der Directory [Benutzername.BATCH]

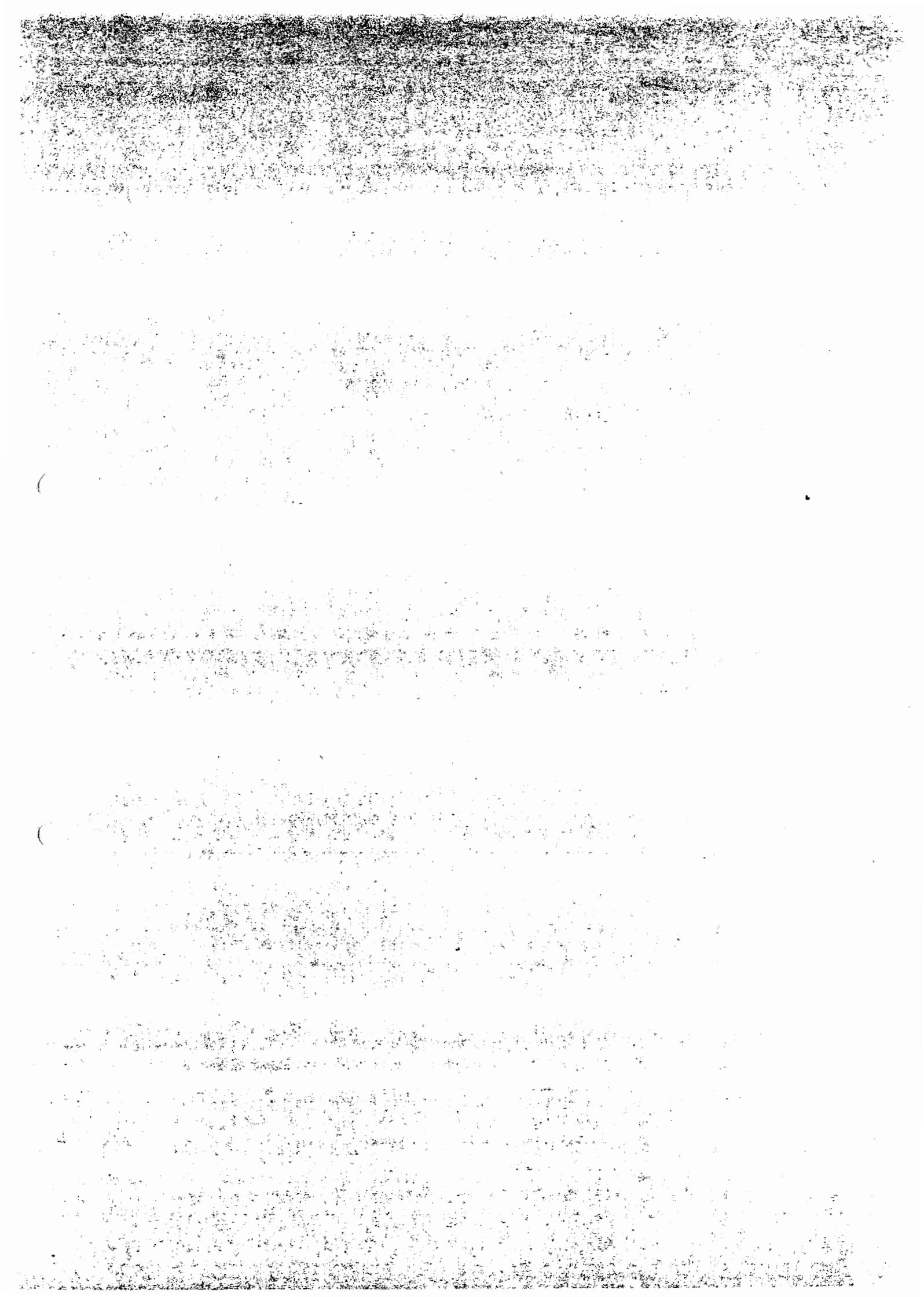
Dateiname der Prozedur: [EMOS.TOOLS]RUNEMOS2.COM

Aufruf der Prozedur: @ [EMOS.TOOLS]RUNEMOS2.COM P1 P2 P3 P4 P5

Parameter im Aufruf:

- P1= Benutzername (z.B. EMOS)
- P2= Name des Files mit dem Hauptprogramm (default= EMOS2)
- P3= vollstaendige Filespezifikation des Job-Input-Files
- P4= vollstaendige Filespezifikation des Ergebnis-Files (default= ['P1'.BATCH]Job-Input-Filename.OUT)
- P5= Angabe, ob Ausdruck des Ergebnis- und LOG-Files erfolgen soll (J/N)

benoetigte Prozeduren: BRUNEMOS2.COM (Batch-Prozedur)
LOGEMOS2.COM



Teil D: EINGABEBESCHREIBUNG

D 1 UEBERSICHT UEBER DAS JOB-INPUT-FILE

Das Job-Input-File enthaelt Modelleingangsdaten, die zur Steuerung des Programmablaufs, des Zeitrahmens und des Umfangs des Ergebnisausdruckes notwendig sind. Daneben kann das Job-Input-File barrierenspezifische Daten enthalten. Die Zuordnung der Datenfiles fuer einen Rechenlauf erfolgt ueber die Angabe von Satznummern. Jede Satznummer ist im INDEX-File mit einem speziellen Datenfilenamen verknuepft (siehe Kap. B 1.2).

Im Einzelnen sind im Job-Input-File folgende Angaben enthalten:

1. Benutzername
2. Start-Zeile
 - Modulname (=REPOS)
 - Satznummern der Daten-Files aus der Datenbibliothek, in denen sich die Eingangsdaten fuer die aktuelle Rechnung befinden. Ist die Satznummer fuer die barrierenspezifischen Daten gleich Null gesetzt, so erfolgt das Einlesen der barrierenspezifischen Daten nur aus dem Job-Input-File.
3. Kommentarzeile mit beliebigem Inhalt zur Charakterisierung des Rechenlaufes (erscheint als Kommentar im Ausdruck)
4. Steuerparameter zur Datenausgabe
 - Auswahl von Barrieren fuer den Ausdruck von Rechenergebnissen
 - Auswahl von Nukliden " " " " " "
5. Steuerparameter zur Ausgabe von Eingabedaten
 - Ausdruck von nuklidspezifischen, elementspezifischen, abfallspezifischen und barrierenspezifischen Daten
 - Aktivitaetsinventar des Barrierensystems
 - Kontrollausdruck der Anfangsbelegung diverser Speicherbereiche
6. Steuerparameter fuer den Ausdruck zeitabhaengiger Groessen
 - Festlegung der Zeitschritte im gesamten, bzw. der Zeitschritte in den gewaehlten Zeitbereichen fuer den Ausdruck von Rechenergebnissen
 - Festlegungen zum Inhalt der Zwischenergebnisausgabe
 - Ausdruck der Zeitpunkte, zu denen maximale Elementkonzentrationen und erstmalig Loeslichkeitsgrenzen erreicht werden, in Tabellen am Ende der Rechnung
7. Steuerparameter fuer die Ergebnisausgabe
 - Ausdruck von Freisetzungsmengen in Bq, kg; Ausdruck von relativen Freisetzungsmengen
 - Auswahl von Nukliden innerhalb Th-Reihe, Np-Reihe, U-Reihe und Am-Reihe, fuer die Massensummen berechnet werden sollen
8. Steuerparameter fuer den Ausdruck von Freisetzungsmengen, mittleren Freisetzungsraten fuer ausgewaehlte Zeitintervalle
 - Ausdruck von Freisetzungsmengen in Bq, kg, Sv
 - Ausdruck von relativen Freisetzungsmengen
 - Ausdruck von mittleren Freisetzungsraten in Bq/a, Bq/s, kg/a, kg/s innerhalb der ausgewaehlten Zeitintervalle

- Auswahl der Barriere fuer die die Freisetzungsmengen, bzw. -raten ausgedruckt werden sollen

9. Steuerparameter fuer die Plotausgabe

- noch nicht realisiert -

- 10. Konstanten zum Zeitrahmen und zur Schrittweitensteuerung
 - Zeitpunkt des Loesungszutrittes
 - Dauer des Szenarios
 - Dauer der Betriebsphase
 - Anfangs- und minimale Schrittweite, relative Schrittweitenverkuerzung, bzw. -verlaengerung und weitere Konstanten zur Schrittweitensteuerung
- 11. Eingabe einer Multibarrierenstruktur
 - Name der uebergeordneten Barriere und Liste der Namen der Barrieren, die in die uebergeordnete Barriere liefern
- 12. Globale Barrierendaten
 - diverse Umrechnungsfaktoren
 - diverse andere Konstanten
- 13. Barrierenspezifische Daten

Ist die Satznummer fuer die barrierenspezifischen Daten groesser Null (vgl. Kap. D 2), werden sie aus dem entsprechenden Datenfile eingelesen. Die barrierenspezifischen Daten im Job-Input-File dienen in diesem Fall der Korrektur und Ergaenzung der bereits eingelesenen Daten.
- 14. Spezifizierung von Abfallmixturen

D 2 EINGABE DER DATEN FUER DAS JOB-INPUT-FILE

In diesem Kapitel werden die einzulesenden Daten des Job-Input-Files zeilenweisen beschrieben. Dem Variablennamen folgt jeweils das Format des einzugebenden Zahlenwertes, sowie nachfolgend eine textliche Erlaeuterung der Bedeutung.

Mit Ausnahme der Variablen IDR(2), duerfen alle INTEGER- und REAL-Variablen nur mit einem positiven Wert oder mit Null besetzt werden. Wird die Funktion, die durch eine Variable gesteuert wird, nicht gewünscht, oder ist die Variable zur Zeit ohne Bedeutung, sollte sie grundsaeztlich gleich Null gesetzt werden.

CHARACTER-Groessen muessen innerhalb des Formats rechtsbuendig eingegeben werden.

1. Zeile:

CNAME[A10]

CNAME : Name des Benutzers

Diese Zeile wird fuer den gesamten Programmablauf einmal eingegeben.

Fuer jeden Modul (zur Zeit nur REPOS), der in einem Programmablauf abgearbeitet werden soll, schliessen sich dann die Eingaben an, die in den Unterkapiteln D 2.1 bis D 2.1 beschrieben sind.

Die Angaben zu den Abfallmixturen (siehe D 2.6) werden jedesmal mit der Zeile:

```
***** .....[A10]
```

beendet.

Fuer den Fall, dass weitere Module aufgerufen werden sollen, werden die Eingaben mit dem Unterkapitel D 2.1 fortgesetzt. Andernfalls wird das Job-Input-File mit der Zeile:

```
CNAME .....[A10]
```

```
CNAME = STOP
```

abgeschlossen.

D 2.1 DATENFILES FUER DEN MODUL

Die Datenversorgung fuer die Bearbeitung eines Programmoduls erfolgt sowohl aus dem Job-Input-File als auch durch Zugriff auf verschiedene Datenfiles der Datenbibliothek. Dazu werden nach der Zeile mit dem Modulnamen die sogenannten Satznummern derjenigen Datenfiles eingegeben, auf die das Programm nachfolgend zugreifen soll (siehe Beschreibung: INDEX-File Kap. B 1.2).

1. Zeile:

```
CNAME (IST(K), K=1,....,6) .....[A10,6I5]
```

CNAME : Filename des Modul

1 (<= IST(K) <= 99

IST(1) : Satznummer des Datenfiles mit den nuklidspezifischen Daten

IST(2) : Satznummer des Datenfiles mit den elementspezifischen Daten. Falls keine elementspezifischen Daten benoetigt werden: IST(2) = 0

IST(3) : Satznummer des Datenfiles mit den abfallspezifischen Daten

IST(4) : Satznummer des Datenfiles mit den barrierenspezifischen Daten. Falls keine barrierenspezifischen Daten gelesen werden sollen: IST(4) = 0

IST(5) : - zur Zeit ohne Bedeutung -

IST(6) : - zur Zeit ohne Bedeutung -

Die anzugebenen Satznummern sind im INDEX-File (Kap. B 1.2) mit dem entsprechenden Filenamen des Datenfiles verknuepft.

3. Zeile:

Kommentarzeile, erscheint als Kopfzeile im Ergebnisausdruck

```
CTEXTI .....[A80]
```

D 2.2 STEUERPARAMETER ZUR DATENAUSGABE

Die Datenausgabe in EMOS2 ist in drei Teile gegliedert:

- Ausgabe der Eingabedaten
- Ausgabe von Zwischenergebnissen waehrend der zeitabhaengigen Rechnung
- Ergebnisausgabe am Ende der Rechnung

Der Ausgabeumfang kann fuer alle Teile durch die Beschraenkung auf einzelne Barrieren und Nuklide reduziert werden.

D 2.2.1 AUSWAHL VON BARRIEREN UND NUKIDEN

1. Zeile:

```
***AUSWAHL (IAU(K), K=1,...,10) .....[A10,10I5]
IAU(1) (<= 0 ODER IAU(1) >= 15 : Ausdruck fuer alle Barrieren
1 (<= IAU(1) <= 14 : Ausdruck fuer IAU(1) ausgewaehlte Barrieren
IAU(2) (<= 0 ODER IAU(2) >= 11 : Ausdruck fuer alle Nuklide
1 (<= IAU(2) <= 10 : Ausdruck fuer IAU(2) ausgewaehlte Nuklide
IAU(3) - IAU(10) : - ohne Bedeutung -
```

2. und 3. Zeile:

```
(CBAU(K), K=1,...,7) .....[7A10]
(CBAU(K), K=8,...,14) .....[7A10]
Fuer K=1,...,14:
CBAU(K) : Name der ausgewaehlten Barriere
```

4. und 5. Zeile:

```
(CNAU(K), JAU(K), K=1,...,5) .....[5(A10,1X,I3)]
(CNAU(K), JAU(K), K=6,...,10) .....[5(A10,1X,I3)]
Fuer K=1,...,10:
CNAU(K) : Name des ausgewaehlten Nuklids
JAU(K) : Atomgewicht des ausgewaehlten Nuklids
```

Die Zeilen 2 bis 5 muessen vorhanden sein. Bei einem entsprechenden Wert von IAU(1) bzw. IAU(2) (z.B. IAU(1)=0) ist ihr Inhalt ohne Bedeutung.

Ist IAU(1)(<0 oder IAU(1)>14 angegeben, wird IAU(1) im Programm gleich 0 gesetzt. Ist IAU(2)(<0 oder IAU(2)>10 angegeben, wird IAU(2) im Programm gleich 0 gesetzt.

D 2.2.2 STEUERUNG DER AUSGABE DER EINGABEDATEN

1. Zeile:

*****INPUT (IIN(K), K=1,...,10)[A10,10I5]

Fuer K=1,...,10:

IIN(K) = 0 : Kein Ausdruck

IIN(1) > 0 : Ausdruck nuklidspezifischer Daten

IIN(2) > 0 : Ausdruck elementspezifischer Daten

IIN(3) > 0 : Ausdruck abfallspezifischer Daten

IIN(4) > 0 : Ausdruck der allgemeinen Abfalldaten, ausgewaehlter
Massensummen, sowie der Waermeleistung, des Gefaehr-
dungspotentials und der Aktinidenmassen.
Ausdruck des Aktivitaetsinventars des Barrierensystems

IIN(4) = 1 : Ausdruck Aktivitaetsinventar in Bq
= 2 : " " " in kg
= 3 : " " " in Bq und kg

IIN(5) > 0 : Ausdruck barrierenspezifischer Daten

IIN(6) > 0 : Kontrollausdruck der Anfangsbelegung folgender
Speicherbereiche:

- Ausdruck der Nuklidnamen mit den Nummern der
Position des zugehoerigen Elements im Vektor CEY;
- Ausdruck der kodierten Strukturmatrix ISY;
- Ausdruck der Barrierenmatrix CB1X sowie der
Vektoren IB1Y und ABD;
- Ausdruck der Eingangsvektoren IB2X und RB2X fuer
jede Barriere;
- Ausdruck des Vektors RA1Y mit den Aktivitaetsinven-
taren und des Vektors RA2Y mit den allgemeinen
Abfalldaten fuer jede Barriere

IIN(7) - IIN(10): - nicht benutzt -

Die auszudruckenden Eingangsdaten beziehen sich auf den Zeitpunkt der
Einlagerung. Bei IIN(4) > 0 wird das Aktivitaetsinventar ausgegeben, das
in der Subroutine BARINV (siehe A 2.4.7) berechnet wird. Die Massen-
summen, die Waermeleistung, das Gefaehrdungspotential und die Aktiniden-
massen werden jeweils bezogen auf dieses Inventar ermittelt. Die Auswahl
der Nuklide, die einen Beitrag zu den Massensummen liefern, erfolgt nach
der Zeile, die mit ****OUTPUT beginnt. Das Gefaehrdungspotential wird in
der derzeitigen Programmversion nicht korrekt berechnet, da sich die
Dimension der Dosiskonversionsfaktoren geaendert hat.

D 2.2.3 STEUERUNG DER AUSGABE VON ZWISCHENERGEBNISSEN

1. Zeile:

*****DRUCK (IDR(K), K=1,...,10)[A10,10I5]

IDR(1) = 1 : Ausdruck der Vektoren RK1Y und RK2Y mit kumulierten
Groessen

IDR(2) = 0 oder IDR(2) >= 6 : Kein Ausdruck von Zwischenergebnissen

IDR(2) < 0 : Ausdruck fuer jeden IDR(2)-ten Zeitschritt

1 <= IDR(2) <= 5 : Ausdruck erfolgt in IDR(2)-Zeitbereichen
(Eingabe der Zeitbereiche: 2. Zeile)

IDR(3) > 0 : Ausdruck von Zeitpunkten, zu denen maximale Element-
konzentrationen und erstmalig Loeslichkeitsgrenzen
erreicht werden

IDR(4) = 1 : - z. Zt. ohne Bedeutung -

IDR(5) : Auswahl des IDR(5)-ten Nuklids fuer den Ausdruck
interessierender barrierenspezifischer Uebergabedaten

IDR(6) = 1 : Ausdruck der Felder RB3Y aus /WZBY/,
RA1Y, RA2Y aus /WZUY/,
RU1Y, RU2Y, RU3Y aus /WZUY/,
sowie der Vektoren EVN, EVS vor und nach dem BMOD-
Aufruf bei jedem Zeitschritte

IDR(7) = 1 : Ausdruck des Aktivitaetsinventars

IDR(8) = 1 : Ausdruck von RB4Y (KNY,1,KBY)

IDR(9) = 1 : " " " (KNY,2,KBY)

IDR(10)= 1 : " " " (KNY,3,KBY)

Ist IDR(1) < 0 oder IDR(1) > 1 angegeben, wird IDR(1) im Programm
gleich 0 gesetzt.

Ist IDR(2) > 5 angegeben, wird IDR(2) im Programm gleich 0 gesetzt.

Ist IDR(5) < 0 angegeben, wird IDR(5) im Programm gleich 0 gesetzt.

Im Gegensatz zu allen anderen Optionen erfolgt der Ausdruck bei
IDR(6) = 1 grundsaeztlich fuer alle Zeitschritte, alle Barrieren
und alle Nuklide. Die Ausgabe IDR(6) = 1 ist daher sehr speicher-
platz-intensiv und sollte nur vorgenommen werden, wenn vorher sicher-
gestellt ist, dass ausreichend Speicherplatz auf der Benutzerplatte
vorhanden ist.

2. Zeile:

(ZDR(K), NDR(K), K=1,...,5)[5(E10.3, I5)]

Fuer K=1,...,IDR(2) !

ZDR(K) : Oberes Ende des Zeitbereichs K in Jahren

NDR(K) : Ausdruck im Bereich K fuer jeden NDR(K)-ten Zeitschritt

Es muss ZDR(K) < ZDR(K+1) fuer K=1,...,IDR(2)-1 gelten.

ZDR(IDR(2)) wird vom Programm auf 1.E20 gesetzt. Die Zeile muss auf jeden Fall vorhanden sein, auch wenn IDR(2) < 1 oder IDR(2) > 5 angegeben wurde; die Werte fuer ZDR(K) sind in diesem Fall ohne Bedeutung.

D 2.2.4 STEUERUNG DER ERGEBNISAUSGABE AM ENDE DER RECHNUNG

Die Ergebnisausgabe am Ende der Rechnung ist in zwei Teile gegliedert:

- Ausgabe der ueber die Szenariendauer kumulierten Freisetzungsmengen
- Ausgabe der kumulierten Freisetzungsmengen und mittleren Freisetzungsraten fuer bestimmte Zeitbereiche

Waehrend der erste Teil entsprechend der Auswahl von Barrieren und Nukliden ausgegeben wird, bezieht sich der zweite Teil nur auf eine anzugebende Barriere, jedoch grundsaeztlich auf alle Nuklide.

A.) Steuerung der Ausgabe kumulierter Freisetzungsmengen

1. Zeile:

****OUTPUT (IOU(K),K=1,...,6)[10X,6I5]

Fuer K=1,...,6:

IOU(K) = 0 : kein Ausdruck

IOU(1) = 1 : Ausdruck von kumulierten Freisetzungsmengen in Bq

IOU(2) = 1 : Ausdruck von kumulierten Freisetzungsmengen in kg und Massensummen fuer die ausgewaehlten Nuklide der Th-, Np-, U- und Am-Zerfallsreihen

IOU(3) = 1 : Ausdruck von kumulierten relativen (auf das eingelagerte Inventar bezogenen) Freisetzungsmengen

IOU(4) - IOU(6) : - z. Zt ohne Bedeutung -

2. Zeile:

(CNTHR(K), ITHR(K), K=1,...,5)[5(A10,1X,I3)]

Fuer K=1,...,5:

CNTHR(K), ITHR(K) : Name und Atomgewicht des ausgewaehlten Nuklids zur Berechnung der Massensumme der Th-Reihe

3. Zeile:

(CNNPR(K), INPR(K), K=1,...,5)[5(A10,1X,I3)]

Fuer K=1,...,5:

CNNPR(K), INPR(K) : Name und Atomgewicht des ausgewaehlten Nuklids zur Berechnung der Massensumme der Np-Reihe

4. Zeile:

(CNUR(K), IUR(K), K=1,...,5)[5(A10,1X,I3)]

Fuer K=1,...,5:

CNUR(K), IUR(K) : Name und Atomgewicht des ausgewaehlten Nuklids
zur Berechnung der Massensumme der U-Reihe

5. Zeile:

(CNAMR(K), IAMR(K), K=1,...,5)[5(A10,1X,I3)]

Fuer K=1,...,5:

CNAMR(K), IAMR(K) : Name und Atomgewicht des ausgewaehlten Nuklids
zur Berechnung der Massensumme der Am-Reihe

Die Massensumme der ausgewaehlten Nuklide einer Zerfallsreihe ist die ueber die Szenariendauer kumulierte, freigesetzte Nuklidmasse in kg

B.) Steuerung des Ausdrucks von Freisetzungsmengen und mittleren Freisetzungsraten fuer ausgewaehlte Zeitintervalle

1. Zeile:

****MENGEN (IME(K), K=1,...,10)[A10,10I5]

Fuer K=1,...,10:

IME(K) = 0 : Kein Ausdruck

IME(1) > 0 : Ausdruck von Freisetzungsmengen in Bq, die bis
ZEFT(K), K=1,...,5 (siehe 2. Zeile) kumuliert sind

IME(2) > 0 : Ausdruck von Freisetzungsmengen in kg, die bis
ZEFT(K), K=1,...,5 (siehe 2. Zeile) kumuliert sind

IME(3) : - z. Zt. ohne Bedeutung -

IME(4) > 0 : Ausdruck von auf das eingelagerte Inventar bezogene
Freisetzungsmengen, die bis ZEFT(K), K=1,...,5
(siehe 2. Zeile) kumuliert sind

IME(5) : - z. Zt. ohne Bedeutung -

IME(6) = 1 : Ausdruck von mittleren Freisetzungsraten in Bq/a in
den ausgewaehlten Zeitbereichen
ZEFT(K) - ZEFT(K-1), K=1,...,5 (siehe 2. Zeile)

IME(6) > 1 : Ausdruck von mittleren Freisetzungsraten in Bq/s in
den ausgewaehlten Zeitbereichen
ZEFT(K) - ZEFT(K-1), K=1,...,5 (siehe 2. Zeile)

IME(7) = 1 : Ausdruck von mittleren Freisetzungsraten in kg/a in
den ausgewaehlten Zeitbereichen
ZEFT(K) - ZEFT(K-1), K=1,...,5 (siehe 2. Zeile)

IME(7) > 1 : Ausdruck von mittleren Freisetzungsraten in kg/s in
den ausgewaehlten Zeitbereichen
ZEFT(K) - ZEFT(K-1), K=1,...,5 (siehe 2. Zeile)

IME(8) bis IME(10) : - z. Zt. nicht benutzt -

2. Zeile:

CNAZEF (ZEFT(K), K=1,...,5)[A10,5E12]

CNAZEF : Name der Barriere, fuer die Freisetzungsraten bzw. Freisetzungsmengen ausgedruckt werden sollen

Fuer K=1,...,5:

ZEFT(K) : Obere Grenzen der Zeitintervalle in Jahren, bezogen auf den Beginn der Freisetzung

Es muss ZEFT(K) < ZEFT(K+1) fuer K=1,...,4 gelten. ZEFT(5) wird im Programm gleich 1.0E+20 gesetzt.

Zeile 2 muss auf jeden Fall eingegeben werden, auch wenn IME(K) = 0 fuer alle K=1,...,10 gewaehlt wurde; die eingegebenen Werte sind in diesem Fall ohne Bedeutung.

D 2.2.5 STEUERUNG DER PLOTAUSGABE

Die naechsten Zeilen sind fuer die Eingabe von Daten vorgesehen, die die Ausgabe von Plots beeinflussen. Alle Zeilen muessen vorhanden sein, obwohl eine in das Programm integrierte Plotausgabe derzeit noch nicht realisiert ist. Die eingegebenen Werte sind somit ohne Bedeutung.

1. Zeile:

*****PLOTS (IPL(K), K=1,...,10)[A10,10I5]

2. Zeile:

(CBPL(K), K=1,...,7)[7A10]

3. Zeile:

(CBPL(K), K=8,...,14)[7A10]

4. Zeile:

(CNPL(K),JPL(K), K=1,...,5)[5(A10,1X,I3)]

5. Zeile:

(CNPL(K),JPL(K), K=6,...,10)[5(A10,1X,I3)]

D 2.3 PARAMETER ZUM ZEITRAHMEN UND ZUR SCHRITTWEITENSTEUERUNG

Der zeitliche Bezugspunkt fuer die Rechnung ist der Beginn der Nachbetriebsphase. Dieser faellt mit dem Ende der Betriebsphase zusammen. Alle Zeitangaben beziehen sich auf diesen Zeitpunkt, wenn nichts anderes vermerkt ist. Mit der Dauer des Szenarios ist die Zeitdauer gemeint, deren Verlauf - eventuell in Abweichung von der tatsaechlichen Szenariodauer - in der Rechnung betrachtet wird. Alle Zeitangaben in Jahren ([a])

1. Zeile:

****ZEITEN, TL, DASZ, TB[A10,3E12.5]

TL : Zeitpunkt des Loesungszutritts

DASZ : Dauer des Szenarios

TB : Dauer der Betriebsphase

2. Zeile:

DT1, EPSI, DDTU, DDTO, ANTKM[10X,5E12.5]

DT1 : Anfangs- und minimale Schrittweite

EPSI : Konstante zur Berechnung des Bereichs einer relativen Stromaenderung in einem Zeitintervall, die zu keiner Aenderung der Schrittweite fuehrt

DDTU : Relative Schrittweitenverkuerzung

DDTO : Relative Schrittweitenvergroesserung

ANTKM : Minimaler Anteil des aktuellen Freisetzungstromes an der bereits kumulierten Freisetzungsmenge, damit die relative Stromaenderung Einfluss auf die Schrittweitensteuerung hat

D 2.4 STRUKTUR DES BARRIERENSYSTEMS

1. Zeile:

**STRUKTUR[A10]

2. Zeile und folgende:

CSX(1), (CSX(K), K=2,...,8)[8A10]

CSX(1) : Name einer Barriere, die Radionuklide aufnimmt

Fuer K=2,...,8:

CSX(K) : Name der Barriere, die ihre Radionuklide an die Barriere CSX(1) weiterleitet

Falls CSX = "KOM: ", wird die Zeile als Kommentar interpretiert. Die Zeile 2 muss analog fuer jeden Knoten des Multibarrierensystems eingegeben werden.

D 2.5 BARRIERENDATEN

D 2.5.1 Globale Barrierendaten

1. Zeile:

****GLOBAL[A10]

2. Zeile:

NG, (G(K), K=1,...,5)[I10,5E12.5]

NG : Anzahl der globalen Barrierendaten

Fuer K=1,...,5:

G(K) : hier ohne Bedeutung

3. Zeile und folgende:

(G(K), K=6,...,NG)[10X,5E12.5/10X,5E12.5 usw.]

G(6) ... G(50) : hier ohne Bedeutung

G(51) : Faktor zur Variation bei LG (Loeslichkeitsgrenze)

G(52) : Faktor zur Variation bei KL (Sorption)

G(53) : nicht benutzt

G(54) : Faktor zur Variation der Mobilisierungsdauern

G(55) : nicht benutzt

D 2.5.2 Spezifische Barrierendaten

1. Zeile:

*BARRIEREN[A10]

Die weiteren Zeilen entsprechen der Beschreibung der barrierenspezifischen Daten in Kap. B 1.3.4

2. Zeile:

Allgemeine Angaben zur Barriere:

CB1X(1), CB1X(2), CB1X(3), CB1X(4), IB1X(1), IB1X(2)[4A10,2I5]

CB1X(1): Name der Barriere

CB1X(2): Name des Barrierenmodells fuer die Barriere

- CB1X(3): Name des Abfalls oder der Abfallmixture, falls die Barriere ein Abfallgebinde ist
- CB1X(4): Name einer anderen Barriere, von der die Eingangsvektoren IB2X und RB2X uebernommen werden (falls erwuenscht)
- IB1X(1): Relative Haeufigkeit dieser Barriere in der naechst-aeusseren Barriere (Wiederholungsfaktor)
- IB1X(2): Anzahl der Einzelabfallgebinde in dem Gebinde CB1X(3)
 =1: Unter CB1X(3) wird der Name eines Einzelabfallgebundes aus den abfallspezifischen Daten angegeben
 >1: Unter CB1X(3) wird der Name eines gemischten Abfallgebundes eingegeben. Die Mischanordnung ist im Job-Input-File festgelegt, Kap D 2.6

3. Zeile:

Forts.: Allgemeine Angaben zur Barriere:

(RB1X(K), K=1,....,5)[10X,5E12.5]

RB1X(1) : Zeitdauer zwischen dem Beginn der Betriebsphase und dem Einlagerungszeitpunkt [a]

RB1X(2) : Rechenergebnis: Beginn des Laugenzuflusses [a]

RB1X(3) : Rechenergebnis: Ende des Laugenzuflusses [a]

RB1X(4) : Rechenergebnis: Ende der Mobilisierung oder Zeitpunkt des Erreichens der Endporositaeet

RB1X(5) : nicht benutzt

RB1X(2), RB1X(3), RB1X(4) muessen bei der Eingabe den Wert Null erhalten !

Falls CB1X(4) nicht " ", werden die weiteren Eingangsdaten vom Gebinde CB1X(4) uebernommen. Das Gebinde CB1X(4) darf an beliebiger Stelle in der Liste der barrierenspezifischen Daten stehen.

4. und 5. Zeile:

Angaben zu dem INTEGER-Eingangsdatenvektor IB2X:

(IB2X(K), K=1 ,....,12)[10X,12I5]

(IB2X(K), K=13,....,24)[10X,12I5]

IB2X(1) : Anzahl der REAL-Eingangsdaten (6. Zeile und folgende)

IB2X(2) : Anzahl der nuklidspezifischen Datensaeetze im Feld RB4Y in /WZBY/

IB2X(3) : NRB3Y in /WZBY/, d.h. Anzahl der barrierenspezifischen, zeitabhaengigen Variablen

IB2X(4) : Anzahl der anderen Uebergangstroeme (Feld RU2Y in /WZUY/)

IB2X(5) : Anzahl der weiteren Uebergabedaten (Feld RU3Y in /WZUY/)

- IB2X(6) : Anzahl der zu druckenden Positionen der anderen Uebergangsstroeme (\leq IB2X(4))
- IB2X(7) : Anzahl der zu druckenden Positionen der weiteren Uebergabedaten (\leq IB2X(5))
- IB2X(8) : - z. Zt. ohne Bedeutung -
- IB2X(9) : - z. Zt. ohne Bedeutung -
- IB2X(10) : - z. Zt. ohne Bedeutung -
- IB2X(11) : - z. Zt. ohne Bedeutung -
- IB2X(12) : - z. Zt. ohne Bedeutung -
- IB2X(13) : - z. Zt. ohne Bedeutung -
- IB2X(14) : Nummer der Zeile der Matrix REY, die die Loeslichkeitsgrenzen enthaelt
- IB2X(15) : Nummer der Zeile der Matrix REY, die die KD-Werte enthaelt
- IB2X(16) : Nummer der Zeile der Matrix REY, die die Mobilisierungsdauern enthaelt
- IB2X(17) : Schalter:
 = 1: wenn es Nuklide in einer Zerfallsreihe gibt, deren zugehoerige Elemente unterschiedliche Mobilisierungsdauern haben
 = 0: sonst
- IB2X(18) bis IB2X(24) : nicht benutzt

6. Zeile und folgende:

Angaben zum REAL-Eingangsdatenvektor RB2X:

(RB2X(K), K=1,...,IB2X(1)) [10X,5E12.5/10X,5E12.5 ... usw.]

RB2X(1), ... ,RB2X(IB2X(1)) :

Angaben sind barrierenmodell-spezifisch und werden im Barrierenmodell erlaeutert

Alle Angaben wiederholen sich analog fuer jede Barriere mit Beginn der 2. Zeile !

D 2.6 SPEZIFIZIERUNG DER ABFALLMIXTUREN

1. Zeile:

***ABF.MIX [A10]

2. Zeile:

CAMZS, RAM1ZS, IAMZS [A10,F10.0,I5]

CAMZS : Name der Abfallmixture

RAMZS : Gesamtzahl der Gebinde des gemischten Abfalls

IAMZS : Anzahl der Einzelabfaelle in der Abfallmixture

3. Zeile und folgende:

(CAM2(K),RAM2(K), K=1,....,IAMZS)[10X,3(A10,F10.0)/
10X,3(A10,F10.0) usw.]

Fuer K=1,....,IAMZS:

CAM2(K) : Name des Einzelabfallgebundes in den abfallspezifischen
Daten

RAM2(K) : Anzahl der Gebinde eines Einzelabfalls

Die Anzahl der Zeilen richtet sich nach der Anzahl der Einzelabfall-
gebunde. Die Reihenfolge der Einzelabfallgebunde ist beliebig.

Die Anzahl der Mixturen ist auf LAM, die Anzahl der Einzelabfaelle in
einer Mixture ist auf LCIA beschraenkt (siehe Kap. A 3.2)