ECONSTOR
Make Your Publications Visible.

A Service of

ZBW Leibniz-Informationszentrum Wirtschaft
Leibniz Information Centre for Economics

Brockhoff, Klaus; Dietrich, Ottokar; Rueck, Günther F.

Working Paper — Digitized Version

DELPHI.F4: Ein Dialogsystem zur Unterstützung von Prognosegruppen unter Benutzung einer Datenbank

Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 31

Provided in Cooperation with:
Christian-Albrechts-University of Kiel, Institute of Business Administration

Suggested Citation: Brockhoff, Klaus; Dietrich, Ottokar; Rueck, Günther F. (1976) : DELPHI.F4: Ein Dialogsystem zur Unterstützung von Prognosegruppen unter Benutzung einer Datenbank, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 31, Universität Kiel, Institut für Betriebswirtschaftslehre, Kiel

This Version is available at:
https://hdl.handle.net/10419/201512

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

WWW.ECONSTOR.EU

Mitglied der Leibniz-Gemeinschaft

Nr. 31

DELPHI.F4

- ein Dialogsystem zur Unterstützung von Prognosegruppen unter Benutzung einer Datenbank

von

K. Brockhoff, O. Dietrich, G.F. Rueck

Copyright

Institut für
Betriebswirtschaftslehre

Februar 1976

Inhaltsverzeichnis

		Seite
1.	Einleitung	1
2.	Ablauf des Dialogs und Arbeitsweise des Dialogsystems DELPHI.F4	3
2.1	Verwendete Symbole im Ablaufdiagramm	3
2.2	Der Ablauf des Dialogsystems	4
3.	Der Ablauf des Datenbankdialogs DATDIA	6
3.1	Allgemeines zur Organisation des Datenbankdialogs	6
3.2	Einzelne Erläuterungen zum Ablaufdiagramm des Datenbankdialogs	7
3.3	Zur Organisation der Informationssuche in der Datenbank	12
4.	Benutzungshinweise für das Dialogsystem	17
4.1	Festlegung der logischen Einheiten und das Starten des Programmsystems	17
4.2	Eingaben der Kontrolle während des Dialogs	18
4.3	Eingaben der Entscheider während des Dialogs	19

Abbildungen 1 bis 14

1. Einleitung

Das Programmpaket DELPHI.F4 ermöglicht einer Gruppe von Teilnehmern (Entscheidern) einen Dialog zur Entwicklung von Antworten auf Prognosefragen nach dem Prinzip der Delphi-Methode. Es erlaubt dabei den Rückgriff der Entscheider auf eine Datenbank zu dem jeweiligen Problembereich. Entwickelt wurde es aber nicht für diese Aufgabe, sondern um einen unterstützenden Rahmen für die experimentelle Prüfung der Leistung von Prognosegruppen und ihres Informationsverhaltens abzugeben.

Das System ist eine Weiterentwicklung des Programms DIALOG.F4, welches allein zur Abwicklung eines Dialogs nach dem Prinzip der Delphi-Methode zwischen einem Kontrollterminal und den Entscheidern, sowie im beschränkten Maße auch zwischen den Entscheidern, gestattet.[1] Auswertungen der Experimente, die zur Überprüfung verschiedener Einflußfaktoren auf die Effizienz von Prognosegruppen im Jahre 1973 durchgeführt wurden, sind an anderer Stelle dokumentiert.[2]

Die bedeutendste Erweiterung gegenüber der bisherigen Situation besteht in DELPHI.F4 darin, daß dem einzelnen Entscheider nunmehr ein Dialog mit einer Datenbank ermöglicht wird, durch den er potentiell Informationen abrufen kann, die ihn zu einer

---

1 Kaerger, D., Das Dialog-System DIALOG.F.4 für Delphigruppen, Manuskripte aus dem Seminar für Betriebswirtschaftslehre der Universität Kiel, Nr. 20, November 1974.

2 Brockhoff, K., The Performance of Forecasting Groups in Computer Dialogue and Face-to-face Discussion, in: H. Linstone, M. Turoff, Hrsg., The Delphi-Method, Techniques and Applications; Reading/Mass. 1975, S. 291-321.

höheren Leistung bei der Abgabe von Urteilen über künftige Entwicklungen in bestimmten Bereichen befähigen sollen.

Die im folgenden beschriebenen Programme sind in MACRO und FORTRAN IV geschrieben. Sie werden auf einer dialogfähigen Rechenanlage im time-sharing-Betrieb vom Typ PDP 10 eingesetzt. Das Betriebssystem dieser Anlage erlaubt für jeden Benutzer die Zuweisung von maximal 16 logischen Einheiten (Input-Output-Einheiten, wie Platten oder Terminals). Die für die Eingabe benötigten Platten, die für die Durchführung der Experimente benötigte Kontrollstation und die für die Experimente in verschiedener Form erstellten Protokolle belegen 7 Einheiten,so daß zur Zeit maximal 8 Teilnehmer in einer Prognosegruppe zusammenarbeiten können. Durch Verzicht auf einige der aus der Experiment-Situation begründeten Auswertungen könnte unter dem vorliegenden Betriebssystem die Zahl der Teilnehmer auf 11 erhöht werden.

Im folgenden wird der Ablauf des Dialogs anhand eines Ablaufschemas dargestellt und der Aufbau des System DELPHI.F4 beschrieben. Es folgen die spezifischen Benutzungsanleitungen. Danach erfolgt eine Beschreibung des Aufbaus der Datenbank und der Zugriffsmöglichkeiten.[3]

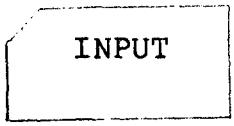
---

[3] Die Erweiterungen des Dialogprogramms DIALOG.F4 und wesentliche Teile des Datenbankdialogs hat Herr Dipl.-Kfm. O. Dietrich vorgenommen. Herr stud.rer.nat.G.F. Rueck hat selbständig schwierige Aufgaben beim Aufbau der Datenbank und bei der Gestaltung des Datenbankdialogs, unter Einschluß der flexiblen Form der graphischen Darstellungen, durchgeführt.

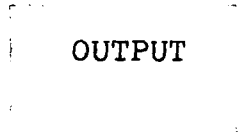
2. Ablauf des Dialogs und Arbeitsweise des Dialogsystems DELPHI.F4

2.1 Verwendete Symbole im Ablaufdiagramm

Bevor auf das Ablaufdiagramm näher eingegangen wird, seien kurz die im Diagramm verwendeten Symbole erläutert.



Das Lochkartensymbol kennzeichnet sowohl eine Eingabe des Experimentators oder der Kontrolle als auch Eingaben der Entscheider von den verschiedenen Bildschirm-Terminals. Diese sind mit Tastatur ausgerüstet.



Dieses Symbol kennzeichnet die Ausgabe von alphanumerischem Text, Graphiken oder Tabellen an den Terminals der Entscheider oder beim Experimentator. Parallel erfolgen ausführliche oder zusammengefaßte Ausgaben auf Plattenbereiche, die allein zur Auswertung der Versuche angelegt werden. Soweit der Ausgabeort nicht besonders gekennzeichnet ist, wird damit eine Ausgabe an allen Bildschirmgeräten gleichzeitig gemeint.

Aufruf eines wichtigen Unterprogramms. In der folgenden Darstellung wird auf die explizite Erwähnung einiger kleinerer Subroutinen verzichtet.

Marke für Sprünge zum Anfang einer Dialogstufe.

Marke für Sprünge in einer Dialogstufe.

Weitere im Ablaufdiagramm erscheinende Symbole haben die übliche Bedeutung.[4]

[4] Nach DIN 66001

2.2 Der Ablauf des Dialogsystems

Nach Zuweisung der logischen Einheiten und dem Start des Programms erhält der Experimentator am Kontrollterminal zunächst einen Überblick über die dem Programm angeschlossenen Peripheriegeräte. Sollten an dieser Stelle Probleme auftreten, insbesondere eine für die Durchführung der Versuche logische Einheit fehlen, so ist es möglich, durch einen einfachen Befehl wiederum Monitor-Level zu erreichen, um eine Korrektur vorzunehmen und das Programm neu zu starten.

Die Eingliederung des Datenbankdialogs in das bestehende Programm macht anschließend eine Initialisierung der für die Datenbank angelegten Dateien bzw. des direkten Zugriffs zu ihnen notwendig.[5] Die Initialisierung läuft in einem kurzen Dialog zwischen dem Kontrollterminal und dem Programm ab. Dabei sind Dateinummer (I) und Name der jeweiligen Dateien (File) anzugeben. Das Programm sieht zur Zeit zwei Dateien vor, die einerseits Erläuterungstexte zu den später von den Benutzern unter Kurzbuchstaben abgefragten Datenreihen enthalten, andererseits diese Datenreihen selbst. Diese Organisation soll lediglich die Flexibilität des Programms zeigen, es ist möglich, auch mehr Dateien zu betreiben, wobei allerdings die oben erläuterte Obergrenze für die Zahl der logischen Einheiten je Programm dann Umdispositionen erfordern würde.

Damit ist die erste Zeile des Ablaufschemas in Abb. 1 erläutert.[6]

Auf den Bildschirmen der Teilnehmer erfolgt nun eine Begrüßung und die Aufforderung, sich gegenüber dem System mit einem Namen zu identifizieren. Anschließend wird von der Kontrolle

[5] Die Realisierung des direkten Zugriffs führte gegenüber DIALOG.F4 zu einer Umorganisation der logischen Einheiten, da das Betriebssystem der PDP 10 direkten Zugriff nur für die Dateien 1-10 vorsieht.

[6] Gegenüber DIALOG.F4 wird auf die Frage nach dem Expertengrad der Teilnehmer verzichtet.

über die Eingabe einer Fragenummer (NF) aus einer eine Reihe von Fragen enthaltenden Datei eine ausgewählt, die den Teilnehmern präsentiert wird.

Unmittelbar darauf wird das Unterprogramm DATDIA.F4 aufgerufen, durch das der Datenbank-Dialog ermöglicht wird (vgl. 3).

Jeder Entscheidende verfügt über eine Liste, die den in der Datenbank abgespeicherten Zeitreihen von Variablen (Informationsarten) Codewörter von je drei Buchstaben Länge zuordnet. Im Datenbankdialog kann er zunächst zu jedem der Codewörter erläuternde Texte erhalten und anschließend auch die Zeitreihen der Informationsarten selbst. Dabei kann eine tabellarische oder eine graphische Darstellung gewählt werden. Außerdem stehen arithmetische Mittel und Streuungen der Zeitreihen auf Anfrage zur Verfügung. Der Datenbankdialog wird, wie unten erläutert wird, individuell beendet. Es folgt die Aufforderung zur Abgabe einer Schätzung auf die eingangs gestellte Frage, die an dieser Stelle wiederholt wird. Sobald hierauf von allen Entscheidern Antworten eingegangen sind, wird geprüft, welche Antworten außerhalb der Quartilsbereiche der dem System übermittelten quantitativen Schätzungen der Entscheider liegen. Die diesen Ergebnissen zugeordneten Benutzer des Systems werden nun gefragt, ob sie über Informationen verfügen, die ihre besondere Position begründen können. Jede darauf eingehende "Zusatzinformation" wird abgespeichert. Nun ist eine Runde (NR als Rundenindex) im Delphi-Dialog beendet. Die Frage wird wiederholt und mit der inzwischen eingegangenen Zusatzinformation den Entscheidern zugestellt. Sie können nun wieder den Datenbankdialog ansprechen und eine neue Antwort zu erarbeiten versuchen. Nach Abschluß von drei Runden (NR=3) wird der Delphi-Dialog abgebrochen und die Ergebnisse werden festgehalten. Es kann nun eine neue Frage bearbeitet werden.

3. Der Ablauf des Datenbankdialogs DATDIA

3.1 Allgemeines zur Organisation des Datenbankdialogs

Der Datenbankdialog ist in mehrere Dialogstufen unterteilt, die innerhalb einer DO-Schleife ablaufen. Jede Dialogstufe besteht aus dem Lesen einer Eingabe, der Ausgabe eines alphanumerischen Textes und einem Hinweis zur Fortsetzung des Dialogs. Innerhalb jeder DO-Schleife wird vom Programm aus für die Teilnehmer der Gruppe eine Art Multiprogramming simuliert. Das Programm ist so gestaltet, daß es immer eine Bearbeitung des Inputs desjenigen Teilnehmers vorsieht, der durch Niederdrücken der "Return"-Taste zu erkennen gegeben hat, daß er eine Eingabe beendete. Dies wird durch ein Unterprogramm (RETURN.MAC in MACRO) geprüft und festgestellt. Damit entspricht die Bearbeitungsdauer des Datenbankdialogs für jeden Entscheider nur derjenigen Zeit, die er selbst für die Bearbeitung der vorliegenden Dialogstufe benötigt.

Die verschiedenen Entscheider werden bei beliebigen Dialogstufen also in derjenigen Reihenfolge bedient, in der sie Dialogstufen durch "Return" abschließen. Die äußere DO-Schleife des Datenbankdialogs ist erst dann erschöpft, wenn alle Entscheider eine Schätzung auf die ihnen gestellte Frage abgegeben haben.

Diese Organisation des Datenbankdialogs ermöglicht es, daß sich die einzelnen Entscheider auf unterschiedlichen Dialogstufen befinden können. Einer der Entscheider kann sich z.B. Erläuterungstexte zu verschiedenen Informationsarten ausgeben lassen, während ein anderer, quasi gleichzeitig, sich eine Graphik ausdrucken läßt oder gerade eine Prognosefrage beantwortet. Die Zahl der von jedem Entscheider durchlaufenen Dialogstufen ist dabei nicht vorgegeben. Er bestimmt diese Zahl individuell, ebenso wie die Verweilzeit auf den einzelnen Dialogstufen. Gegenüber einer sequentiellen Abarbeitung des Dialogs der einzelnen Teilnehmer hintereinander ergibt sich also hier der Vorteil, daß sich die Bearbeitungszeiten

der Entscheider nicht aufaddieren. Ein bestimmter Entscheider braucht also nicht erst so lange zu warten, bis derjenige, dessen Bildschirmgerät in der Reihenfolge der logischen Einheiten für das System vor ihm liegt, seinen Dialog abgeschlossen hat. Bei acht Teilnehmern und einer durchschnittlichen Dialogdauer von 8 Minuten für jeden Entscheider wird damit eine potentielle Wartezeit von 56 Minuten für das nach seiner logischen Einheit letzte Mitglied der Gruppe vermieden. Hier hat jeder Entscheider den Eindruck, sofort mit seiner Informationsnachfrage bedient zu werden.

Nach Abgabe einer Antwort auf die jeweils gestellte Frage können für die Teilnehmer Wartezeiten auftreten. Dies deshalb, weil für die Feststellung der Median- und Quartilswerte, sowie für die Befragung derjenigen Teilnehmer, deren Antworten außerhalb der Quartilswerte liegen, zunächst die Antworten aller Gruppenmitglieder eingegangen sein müssen.

Würde man von diesem Erfordernis absehen und das Dialogsystem zu einem allgemeinen Diskussionssystem ausbauen wollen, so könnten auch an dieser Stelle Wartezeiten vermieden werden, da die Diskussion wiederum individuell weitergeführt werden könnte.

3.2. Einzelne Erläuterungen zum Ablaufdiagramm des Datenbank-Dialogs.

In Abb. 2 wird das Ablaufdiagramm des Datenbankdialogs gegeben. Der Datenbankdialog wird für alle Teilnehmer einer Gruppe gleichzeitig mit der Frage eröffnet, ob zu den Codewörtern der vorliegenden Liste von verfügbaren Informationsarten Erläuterungen erwünscht sind. Sofort anschließend beginnt das obenerwähnte simulierte Multiprogramming.

In der dafür geschaffenen DO-Schleife werden alle Entscheider übersprungen, die entweder die ihnen gestellte Frage bereits beantwortet haben oder noch keine Eingabe abgeschlossen haben. Im jeweils anderen Fall wird für jeden Entscheider die Adresse

der im folgenden zu bearbeitenden Dialogstufe gelesen (ADRESS (II)) und ein Sprung zu dieser Adresse vorgenommen.

Bei jeder Adresse wird zunächst ein Output an den Entscheider produziert, der ihn zu einem Input auffordert. Durch diesen Input verlangt der Entscheider bestimmte Informationen aus der Datenbank. Nach Abgabe dieser Information an den Entscheider erfolgt vor Abschluß der Dialogstufe ein Hinweis an den Entscheider für die Fortsetzung des Dialogs. Gleichzeitig wird die Adresse der nächsten für den Entscheider zu bearbeitenden Dialogstufe gespeichert und über den Punkt "A" der Abbildungen 3 bis 12 zurückgesprungen. Aus den Abb. 4 und 5 entnimmt man beispielhaft, daß keineswegs immer zur jeweils nächsten Adresse gesprungen wird; der Übergang erfolgt vielmehr in Abhängigkeit von der jeweiligen Reaktionsweise des Entscheiders.

Nach Abwicklung einer Dialogstufe für einen Entscheider wird nach der oben beschriebenen Prozedur der jeweils nächste Entscheider bearbeitet.

Der Datenbankdialog endet, wenn alle Entscheider eine Schätzung für die Beantwortung der ihnen gestellten Frage abgegeben haben. Das Ende der Dialogschleife wird in Abb. 12 erkennbar. Für einen bestimmten Entscheider wird ADRESS (II) gleich 15 erreicht.

Im folgenden wird eine Erläuterung zur Vorgehensweise in Dialogstufen gegeben, soweit es sich dabei nicht um routinemäßig abgewickelte Vorgänge handelt.

In der ersten Dialogstufe (J=1) werden die vom I-ten Entscheider aus der ihm vorliegenden Liste entnommenen Codewörter gelesen und die Erläuterungen hierzu nach seinen Wünschen ausgegeben.

Für jedes eingegebene Codewort wird einzeln geprüft, ob dieses in einer entsprechenden Liste abgespeichert ist.[7] Hierzu wird eine Routine "SEARCH" angesprochen. Ist das gewünschte Codewort auf der Liste nicht verzeichnet, so folgt an den Entscheider eine Fehlermeldung. Er hat ein Codewort eingegeben,

---

[7] Vgl. hierzu auch Abschnitt 3.3.

für das keine Konventionen vereinbart sind. Sind für ein bestimmtes Codewort keine Erläuterungen zu diesem Wort auf der Datenbank abgespeichert, so erscheint auf dem Bildschirm des diese Erläuterungen suchenden Entscheiders eine Mitteilung, daß sich zu diesem Codewort noch keine Erläuterungen im Datenspeicher befinden. Diese Maßnahme ist vorgesehen, um einen flexiblen Ausbau der Codewörter und der Datenbank vorzunehmen, auch wenn gleichzeitig das System eingesetzt wird.

Die Fehlermeldungen an den Benutzer sind zweistufig angelegt, d.h. daß bei einer wiederholten fehlerhaften Eingabe in einer Dialogstufe eine ausführlichere Fehlermeldung erfolgt als bei einer erstmals fehlerhaften Eingabe. Zur Korrektur der angemerkten Fehler ist lediglich eine Wiederholung der Eingabe in richtiger Form notwendig.

Wenn das gewünschte Codewort auf der Liste gefunden werden kann und auch Erläuterungen hierzu vorliegen, so übermittelt die Subroutine SEARCH dem aufrufenden Programm die Anfangs- und die Endadresse der zum Codewort eingetragenen Information in der Datenbank. Diese Adressen werden einem Unterprogramm "GET" übergeben. Aufgabe dieser Subroutine ist es, die Erläuterungen zu dem gewünschten Codewort von der Datenbank (einer Datei mit der Bezeichnung als logische Einheit 2) zu lesen und einem zweidimensionalen Variablenfeld zu übergeben. Der Inhalt dieses Variablenfeldes kann dann an den Entscheider ausgegeben werden. Dies erfolgt jeweils für ein Codewort. Wünscht der Entscheider, der anfangs die Erläuterungen zu mehreren Codewörtern erbeten hat, eine Weiterführung der Übermittlung von Codewörtern, so kann er dies durch den Befehl "GO" bewirken. Ein entsprechender Hinweis erscheint am Ende jedes Erläuterungstextes. Natürlich steht es dem Entscheider frei, durch erneute Eingabe die Erläuterungen beliebig oft wiederholen zu lassen.

Die von einem Entscheider erwünschten Erläuterungen werden für ihn unter Angabe der Fragennummer und der Runde, in der sich der Delphi-Dialog befindet, auf einer Datei (logische Einheit 5) protokolliert. Diese Maßnahme dient allein der

Auswertung der Experimente zum Informationsverhalten.

Wenn ein Teilnehmer auf die Erläuterung von Codewörtern verzichtet wird er aufgefordert, die Codewörter für solche Informationsarten einzugeben, zu denen er sich die Ausgabe der zugehörigen Datenreihen wünscht. In der zweiten Dialogstufe (J=2) werden die von einem Teilnehmer gewünschten Informationsarten in der oben beschriebenen Form durch die Routinen SEARCH und GET gelesen. Hat sich der Teilnehmer für die Nachfrage nach Daten entsprechend bestimmten Codewörtern entschieden und sind dabei keine Fehler aufgetreten, so hat er anschließend eine Wahl der Darstellungsform für das von ihm gewünschte Material. Das Material kann in Tabellenform oder in graphischer Form, allerdings nur in punktweiser Darstellung, ausgegeben werden.

Die Bearbeitung und Ausgabe der gewünschten Darstellungsform erfolgt in der dritten Dialogstufe (J=3). Die vom Entscheider eingegebenen Codewörter sind so aufgebaut, daß sie erkennen lassen, ob die zugehörigen Zeitreihen aus Jahresdaten, Quartalsdaten oder Monatsdaten bestehen. Wählt der Entscheider nun die graphische Darstellung, so wird ihm zunächst eine Graphik ausgegeben, die die von ihm gewünschten Jahresdaten enthält. Hat er darüber hinaus über Codewörter Datenreihen angefordert, so kann als weiteres von ihm die Darstellung der Graphik mit Informationsarten angefordert werden, die sich allein auf Quartale bezieht; anschließend kann die Graphik mit den Eintragungen der Monatsdaten angefordert werden.

Die Graphiken selbst werden mit Hilfe einer Subroutine "GRA" erstellt. Sie sind so angelegt, daß sie maximal fünf Informationsarten eines Datentyps (Jahres-, Quartals- oder Monatsdaten) wiedergeben können. Die Datenreihen können dabei unterschiedliche Dimensionierungen aufweisen (z.B. Prozentwerte oder DM-Absolutbeträge). Dies wird jeweils angezeigt und bei der automatischen Ermittlung der geeigneten Unterteilung der Ordinatenabschnitte berücksichtigt. Die Abszisse der Graphik wird für jeden darzustellenden Fall errechnet und

individuell wiedergegeben. So kann berücksichtigt werden, daß die Benennungen auf den jeweiligen Datentyp und den dafür vorliegenden Erhebungszeitraum abzustellen sind.

Wünscht der Entscheider eine tabellarische Darstellung, so wird zunächst geprüft, ob er wieder nach unterschiedlichen Typen der Informationsarten fragt. Ist das der Fall, so erhält er einen warnenden Hinweis, daß innerhalb der Tabelle verschiedene Zeiträume durch die Daten abgedeckt werden. Wird nur ein einheitlicher Typ von Informationsarten verlangt, kann sofort zur Ausgabe der Tabelle durch die Subroutine "TAB" übergegangen werden. Wie aus den Abbildungen 6 und 7 hervorgeht, wird dem Entscheider nach Ausgabe der von ihm gewünschten Darstellungsart für alle von ihm gewünschten Informationsarten Gelegenheit gegeben, eine zunächst nicht gewünschte und deshalb ihm auch nicht gewährte Darstellungsart für dieselben Informationsarten anzufordern.

Nach Abschluß der gewünschten Darstellungen ist dem Entscheider die Möglichkeit gegeben, zu den von ihm angeforderten Informationsarten das arithmetische Mittel und die Streuung zu erfragen.

Die gewählten Darstellungsformen und die eventuell erwünschte Ausgabe des Mittelwerts und der Streuung für jede Informationsart wird jedem Entscheider, jeder Frage und jeder Runde im Delphi-Dialog zugeordnet und in einem besonderen Feld gespeichert und protokolliert.

Nach einer einmaligen Durcharbeitung der vom Entscheider eingangs gewünschten Informationsarten in einer Runde einer Frage erhält er in der Dialogstufe J=9 die Option, entweder den Dialog erneut zu starten oder die ihm gestellte Frage zu beantworten. Wünscht er eine Fortsetzung des Dialogs, wird im Ablaufdiagramm zur Marke "INFO" am Beginn der ersten Dialogstufe (Abb. 3) zurückgesprungen. Wünscht der Entscheider keine Fortsetzung des Dialogs, so kann er in der dritten Runde einer ihm gestellten Frage (NR=3) Hinweise auf Informationsarten geben, die er zur Beantwortung der Frage zusätzlich gewünscht hätte, aber in der Datenbank nicht finden konnte. Diese Hinweise können im alphanumerischen Text, d.h. ohne Benutzung

von Codewörtern, gegeben werden. Auch diese Information wird gelesen (vgl. die Dialogstufe J=10) und auf der logischen Einheit 10 protokolliert.

Nach Eingabe einer quantitativen Antwort auf die dem Entscheider gestellte Frage wird in der Dialogstufe J=11 über die übliche Fehlerabfrage hinaus überprüft, ob die Schätzung innerhalb eines plausiblen Intervalls liegt. Auf diese Weise soll vermieden werden, daß formal richtige aber nicht sinnvolle Angaben die Ergebnisse verzerren. Solche Angaben kommen erfahrungsgemäß insbesondere durch fehlerhafte Setzung des Dezimalpunktes vor und verzerren damit die Eingaben um Zehnerpotenzen. Eine Überwindung der vorgegebenen Intervallgrenzen durch den Entscheider ist dadurch möglich, daß er seine Eingabe in der ursprünglichen Form wiederholt.

Mit dem Erreichen dieses Punktes ist der Datenbankdialog für eine Runde und einem Entscheider abgeschlossen. Es ist dies die Stelle, an der die oben beschriebenen Wartezeiten erstmals auftreten können.

3.3. Zur Organisation der Informationssuche in der Datenbank

Die folgenden Ausführungen zur Organisation der Datenbank geben in zweierlei Hinsicht nur einen Teil der Probleme wieder. Nicht berichtet wird über diejenigen Algorithmen, die zur Pflege des Datenbestandes eingesetzt werden. Diese Funktionen laufen unabhängig und außerhalb der Aufgaben ab, die mit der Integration von Datenbank und Delphi-Dialog geleistet werden sollen. - Unberücksichtigt bleibt auch eine detaillierte Schilderung der Übertragung von sogenannten Teilabschnitten (vgl. Programmteil GET in Abb. 13), worin die Übertragung mit einem Buffersystem vorgenommen wird und die Abfrage nach möglichen Systemfehlern vorkommt. Dies würde den hier angestrebten zusammenfassenden Überblick sprengen.

Das FOTRAN-System der PDP-10 gestattet standardmäßig die Einrichtung von Random-Access-Files auf Platten, die einen wahlfreien Zugriff auf einzelne Blöcke erlauben. Die Länge dieser Blöcke beträgt 128 Worte, von denen zwei Worte vom System für dessen eigene Verwaltung reserviert werden. In

den Files sind die Informationen binär gespeichert, womit geringe Zugriffszeiten ermöglicht werden. Das Lesen eines Blockes vom Datenfile (durch GETBLK) und das Schreiben eines Blocks (durch PUTBLK) berührt Operationen zwischen dem File und einem Buffersystem, nicht aber das eigentliche Zugriffssystem. Dies mag für die Anpassung an andere Betriebssysteme vorteilhaft sein.

Der Ablauf der Informationsauffindung und -übertragung ist in Abb. 13 aufgezeigt. Er wird durch die Subroutinen SEARCH und GET bewerkstelligt. Wir benutzen eine indirekt adressierte Datei.

Die Subroutine SEARCH wird mit der Eingabe eines Schlüsselworts aufgerufen. Im vorliegenden Falle bestehen die Schlüsselworte ausschließlich aus drei Symbolen. Grundsätzlich können Schlüsselworte aus 1 bis 80 Symbolen bestehen. Ein Maschinenwort der PDP-10 (36 Bits) kann bis zu 5 Symbole aufnehmen, so daß die Schlüsselworte eine Länge von 1 bis 16 Maschinenworte belegen. Im vorliegenden Falle wird auch nur ein Maschinenwort belegt. Aus dem eingetragenen Schlüsselwort wird über eine Speicherfunktion (hash function) ein Hashcode errechnet. Als Speicherfunktion wird das Zerlegungsverfahren als Spezialfall des Divisions-Rest-Verfahrens verwendet.[1] Der Wert des Hashcodes liegt im Intervall $[0,\ldots,k,\ldots,$ (Länge der Hashliste - 1)]. Nehmen wir an, er erreiche den Wert k .

Zu diesem Wert gibt die Hashliste eine Adresse ℓ für den Anfang eines Blockes in der blockweise aufgebauten Schlüsselwortliste an. Ist die Adresse $\ell = 0$, so ist ein entsprechendes Schlüsselwort nicht eingetragen und es erfolgt die in Abb. 13 vorgesehene Fehlermeldung.

Wird für $\ell > 0$ in die Schlüsselwortliste gesprungen, so ist zu beachten, daß aus verschiedenen Schlüsselworten derselbe Hashcode errechnet werden kann. Nach dem Prinzip der internen Verkettung sind deshalb "Schlüsselwortteillisten" aus mehreren Blöcken zusammengestellt. In den Schlüsselwortteillisten werden

[1] Vgl. Noltemeier, H., Datenstrukturen und höhere Programmiertechniken, Berlin, New York 1972, S. 60 f.

solche Blöcke zusammengefaßt, die zu Schlüsselworten mit gleichem Hashcode gehören. Eine Schlüsselwortteilliste wird daraufhin durchgesucht, ob das zum errechneten Hashcode gehörende Schlüsselwort darin enthalten ist. Um dies zu ermöglichen, ist jeder Schlüsselwortblock aus wenigstens fünf Worten zusammengesetzt, die die folgende Bedeutung haben:

Wort 1 : Der Nachfolger des m-ten Schlüsselwortblocks in der Schlüsselwortteilliste ℓ wird hier mit seiner Adresse angegeben.

Wort 2 : In diesem Wort wird die Länge des Schlüsselworts in Maschinenworten angegeben. Hier wird regelmäßig mit einer Länge von 1 gearbeitet.

Wort 3 : An dieser Stelle ist das Schlüsselwort enthalten. Grundsätzlich können hier sequentiell mehrere Worte aneinander gekettet sein, so daß sich aus dem Wort 2 ein allgemeineres Indiz für das Ende von Wort 3 ergibt, wenn nicht mit der Länge 1 gerechnet wird. Dies wäre bei den folgenden Erläuterungen auch zu berücksichtigen.

Wort 4 : Dieses Wort enthält die Anfangsadresse (ANF) der zum Schlüsselwort eingetragenen Information. Die Eintragung Null zeigt an, daß zum Schlüsselwort keine Information eingetragen ist. Die Adressen sind zweiteilig angelegt. Die linke Hälfte enthält die physikalische Blocknummer des Files, die rechte Hälfte die Nummer des Wortes in diesem Block.

Wort 5 : Hier wird die Endadresse (END) der zum Schlüsselwort eingetragenen Information geführt. Für die Informationsauffindung ist diese Adresse ohne Belang. Der Aufbau der Endadresse entspricht dem in Wort 4 geschilderten Aufbau der Anfangsadresse.

Enthält die Hashliste eine Adresse ℓ > 0 , so wird das Schlüsselwort direkt mit den Eintragungen in dieser Teilliste verglichen, bis entweder das Schlüsselwort identifiziert wird oder das Ende der Teilliste ohne Erfolg erreicht wird. Wird das

Schlüsselwort identifiziert, so wird dem zugehörigen Schlüsselwortblock die Anfangsadresse (ANF) der zugehörigen Information entnommen. Ist ANF = 0 , so liegen zu dem Schlüsselwort keine Informationen vor. Anderenfalls wird diese Adresse an die Subroutine GET übergeben, die die Übertragung der Information übernimmt. Die Information wird beim Eintrag innerhalb von Blöcken in Teilabschnitte (Sätze) variabler Länge verlegt, die wiederum indirekt verkettet sind. Dies erfolgt aus vier Gründen:

1. Es können maximal 126 Worte in einem Block des Files benutzt werden. Es ist aber nicht ausgeschlossen, daß die Information mehr als 126 Worte benötigt.
2. Es kann vorkommen, daß die Information weniger als 126 Worte in einem Block belegt. Die verbleibenden Restteile von Blöcken können nun mit Information zu anderen Schlüsselwortblöcken belegt werden und damit ausgenutzt werden.
3. Durch Löschen von Informationen auf dem File freigewordene Bereiche können leicht neu belegt werden.
4. Die Zerlegung erlaubt eine leichte Erweiterung und Korrektur der Informationen.

Jeder Teilabschnitt i des Informationsfile besteht aus drei Feldern:

1. dem Anfangswort (AW_i),
2. dem eigentlichen Informationsteil ($DATA_i$),
3. dem Endwort (EW_i).

Auf das Anfangswort wird durch die Adresse ANF aus dem Schlüsselwortblock verwiesen. Die Adresse END aus dem Schlüsselwortblock verweist auf das Endwort des letzten von evtl. mehreren zugehörigen Teilabschnitten im Informationsfile. Unter der Adresse des Anfangsworts ist die Adresse des Endworts im Teilabschnitt verzeichnet, d.h. eine physikalische Wortnummer in dem Block. Im zweiteiligen Endwort steht links die Nummer des Blocks, in dem der logisch folgende Teilabschnitt eingetragen ist und rechts die Nummer des Wortes in diesem Block. (In Abb. 14 wird der Zusammenhang ohne Darstellung der physikalischen Blöcke des Informationsfiles angegeben.) Enthält das Endwort den Wert Null, so ist das Ende der zu einem Schlüsselwort abzuarbeitenden Information erreicht.

Die Übertragung eines Teilabschnittes i (siehe Abb. 13) besteht im wesentlichen aus vier Teilen:

1. Lesen des Blockes, in dem der Teilabschnitt eingetragen ist, in den Buffer. Ist der Block bereits im Buffer (dies kommt vor, wenn sich zwei aufeinanderfolgende Teilabschnitte in einem Block befinden), wird das Lesen unterdrückt;
2. Bestimmen des Anfangs und des Endes des Informationsteils (DATA);
3. Übertragen des Informationsteils;
4. Prüfen, ob Fehler aufgetreten sind. Es sind im wesentlichen folgende Fehler möglich:
   a. Fehler beim Lesen des Blockes;
   b. Der Zielbereich, d.h. der Bereich, in den die Information durch GET übertragen wird, ist zu kurz;
   c. Programmfehler.

Mehrere Tests mit dem vorliegenden System haben gezeigt, daß es sicher arbeitet und leicht auch von Benutzern bedient werden kann, die keinerlei Vorkenntnisse der elektronischen Datenverarbeitung haben. Ein Indiz dafür mag auch darin gesehen werden, daß die Datenbank sehr intensiv nach möglicherweise den Urteilsbildungsprozeß unterstützenden Informationen durchgesucht wurde und verschiedene Darstellungsformen von den Teilnehmern gewünscht wurden.

4. Benutzungshinweise für das Dialogsystem

4.1. Festlegung der logischen Einheiten und das Starten des Programmsystems

Das Dialogsystem wird vom Terminal der Kontrolle aus gestartet. Die vom Dialogsystem benötigten logischen Einheiten werden über das Kontrollterminal durch die im folgenden links aufgeführten Anweisungen an das Programm angeschlossen. Rechts steht jeweils die Funktion der angeschlossenen logischen Einheiten.

÷ AS DSK 1 ↵	- Datendatei	} der Datenbank
÷ AS DSK 2 ↵	- Erläuterungsdatei	} der Datenbank
÷ AS DSK 4 ↵	- Protokoll der gewünschten Daten und Darstellungsformen	
÷ AS DSK 5 ↵	- Protokoll der gewünschten Erläuterungen	
÷ AS DSK 10 ↵	- Dialogprotokoll (Langform)	
÷ AS DSK 30 ↵	- Ergebnisfile (Kurzform)	
÷ AS DSK 31 ↵	- Fragendatei	
÷ AS TTY 9 ↵	- Terminal für Kontrolle	
÷ AS TTYnr 11 ↵	- Terminal für Entscheider 1	
÷ AS TTYnr 12 ↵	- Terminal für Entscheider 2	
÷ AS TTYnr 13 ↵	- Terminal für Entscheider 3	
.	.	.
.	.	.
.	.	.
÷ AS TTYnr 18	- Terminal für Entscheider 8	

"nr" bedeutet hierbei die physikalische Einheit des Terminals. Die physikalische Einheit eines Terminals läßt sich durch folgenden Befehl feststellen:

÷ I ↵ - Der Befehl ↵ bedeutet RETURN.

Bei den Entscheidern folgt der physikalischen Einheit nach einem Blank die Nummer der zugewiesenen logischen Einheit.

Das Dialogprogramm kann auf zwei Arten gestartet werden.
Ist das Programm noch nicht geladen worden, so sind folgende Systembefehle einzugeben:

```
÷ LOAD @BANK ↵
÷ SAVE ↵
÷ START ↵
```

BANK ist ein "Commandfile" mit der Bezeichnung ". CMD ". In diesem Befehlsfile sind alle Namen der Programme aufgelistet, die zum Dialogsystem gehören. Diese Ladeprozedur ist im Rechenzentrum der Universität Kiel nur nach 23.oo Uhr möglich, da das Programm zum Laden mehr als 32 K benötigt. Liegt das Programmsystem bereits geladen im Kernspeicher vor, so kommt es mit 29 K aus und kann auch tagsüber gestartet werden. Durch den zweiten Befehl ("SAVE") wird das geladene Programm im Kernspeicher sichergestellt. Der letzte Befehl läßt das Programm starten.

Liegt ein bereits geladenes Programm vor, so kann es mit folgendem Befehl sofort gestartet werden:

```
. RUN DELPHI.SAV ↵
```

DELPHI ist hierbei der Name des Hauptprogramms.

4.2 Eingaben der Kontrolle während des Dialogs

Die vom Dialogsystem ausgegebenen Aufforderungen werden im folgenden unterstrichen wiedergegeben. Durch die einzelnen Zeichen ('␣') wird angedeutet, wie viele Zeichen das Programm als Eingabe am Terminal erwartet.

UNIT : ␣ ↵ z.B. 1	Nummer der logischen Einheit für die Datenbank.
FILE : ␣␣␣␣␣␣␣␣ ↵ z.B. DATØ1.RAF	Name des Datenfiles, für den der direkte Zugriff initialisiert werden soll.

Für die Initialisierung des direkten Zugriffs sind im Dialogsystem die logischen Einheiten "1" und "2" mit den Datennamen "DATØ1.RAF" und "DATØ2.RAF" vorgesehen. Die logischen Einheiten und die Dateinamen können jedoch ohne weiteres geändert und die Zahl der zu initialisierenden Dateien erhöht werden. Die Initialisierung wird durch die Eingabe Ø abgeschlossen:

UNIT : Ø ↵

TEST.. ↵
␣␣

Zweistellige Gruppennummer eingeben (beliebig, zur Bezeichnung der Experimentgruppe oder des Benutzungsfalles).

TEILNEHMERANZAHL MIT - Ø ODER 1 - EINGEBEN
-I2-I1-..

␣␣b␣

Zweistellig: Gruppengröße (Zahl der Entscheider); einstellig: Spezifikation.

(b=Blank-Zeichen angeben).

NUMMER DER FRAGE EINGEBEN-I3-...

␣␣␣

Dreistellige Fragenummer eingeben. Mit Null "Ø" wird das Programm DELPHI.F4 beendet.

4.3 Eingaben der Entscheider während des Dialogs

Der Dialog mit dem Entscheider ist grundsätzlich so aufgebaut, daß bei fehlerhaften Eingaben der Entscheider eine Meldung mit der Aufforderung erhält, seine Eingabe noch einmal zu überprüfen und zu korrigieren. Die erste Eingabe des Entscheiders ist die Eingabe des Namens. Hierbei werden maximal 30 Zeichen gelesen. Während des Datenbankdialogs können alle auf der Codewortliste verzeichneten Schlüsselworte, zu denen sich der Entscheider Erläuterungen oder Datenreihen wünscht, eingegeben werden. Hierbei werden die einzelnen Codewörter, wenn mehrere (maximal 5 Wörter) eingegeben werden, durch Kommata voneinander getrennt, wie z.B. "PAM, KKM, KKQ". Weiterhin können während des Dialogs vom Entscheider folgende Zeichen eingetippt werden:

GRA - es wird eine graphische Darstellung gewünscht
TAB - es wird eine tabellarische Darstellung gewünscht
AR - es werden Angaben über das arithmetische Mittel und die Streuung der angegebenen Informationsarten gewünscht
GO - der Dialog soll fortgesetzt werden
JA
NEIN

Bei der Beantwortung der Fragen können nur positive Zahlen von Ø.ØØ bis 9999999.99 eingegeben werden, wobei anstelle des Kommas ein Punkt erwartet wird, auch wenn es sich um eine ganze Zahl handelt (z.B. "65":= "65.").

Bei den Fragen, welche zusätzliche Informationen auf der Datenbank zur Beantwortung der Prognosefrage wünschenswert gewesen wären und der Begründung der Abweichung vom Mittelwert, kann der Entscheider max. 1 Zeile bzw. 2 Zeilen eingeben.

Abb.1 Ablaufschema des Dialog-Systems

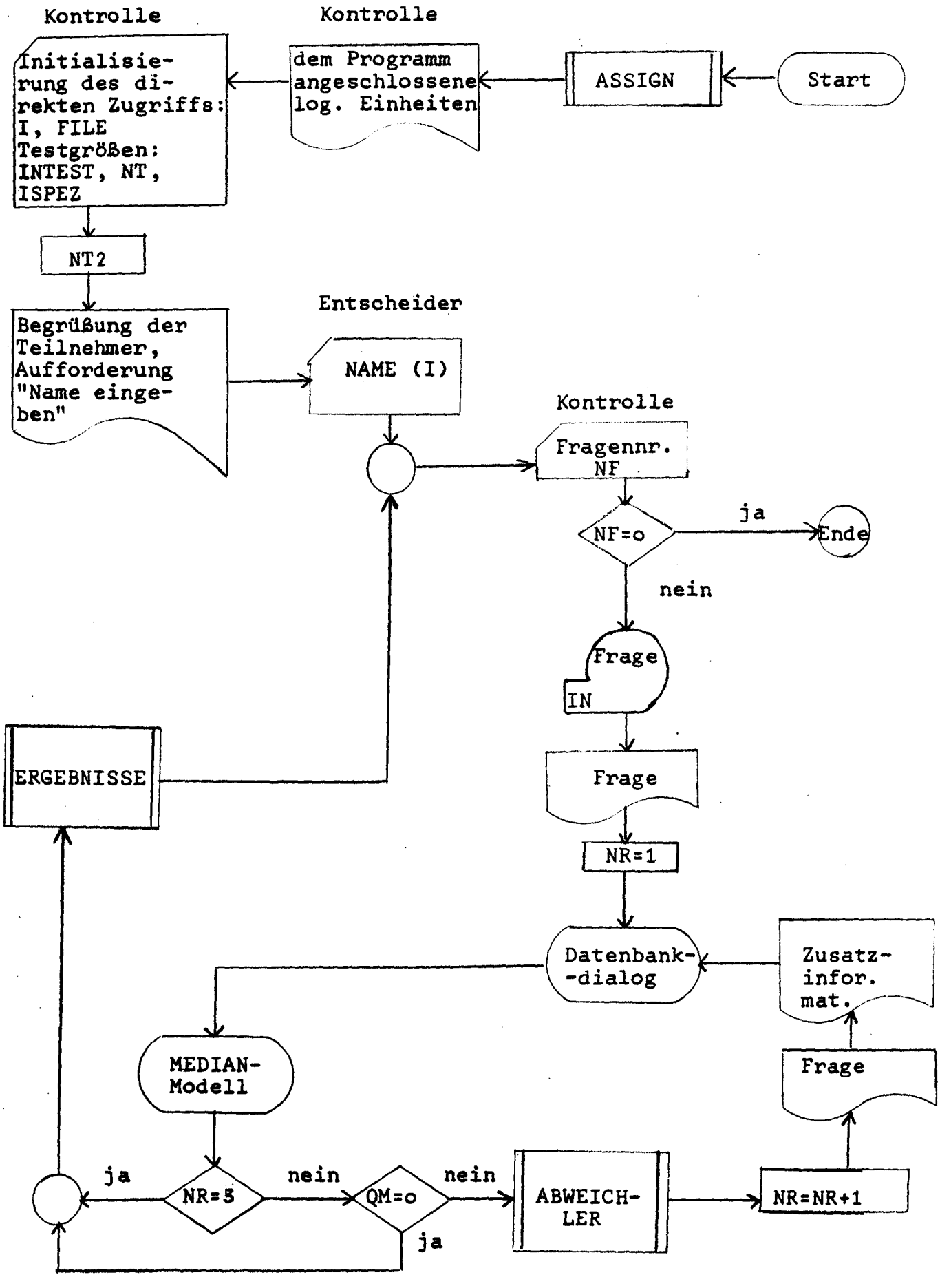


Abb.2 Ablaufdiagramm des Datenbankdialogs

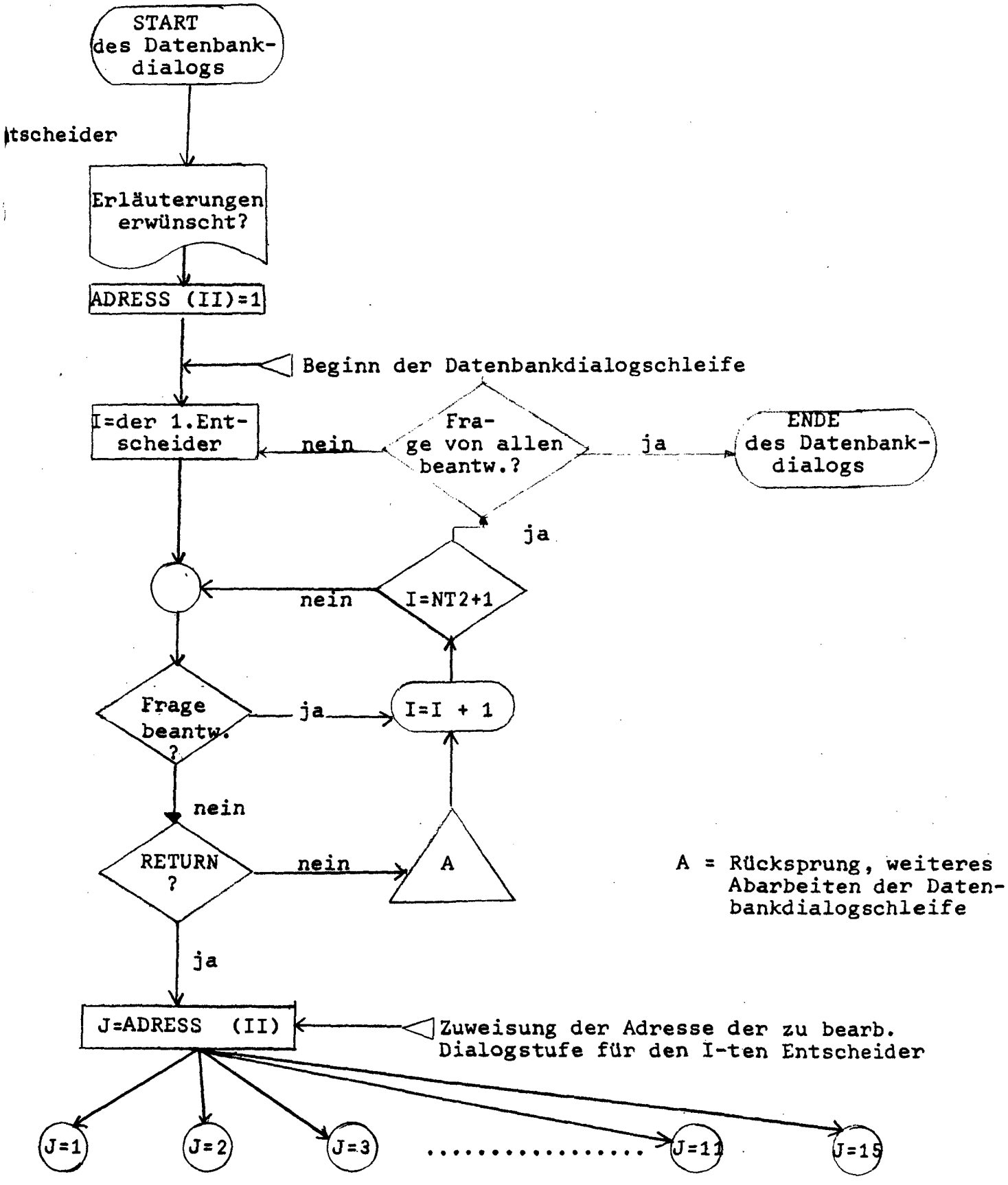


Abb.3 Erste Adresse: Lesen und Ausdrucken der gewünschten Erläuterungen zu den Codewörtern

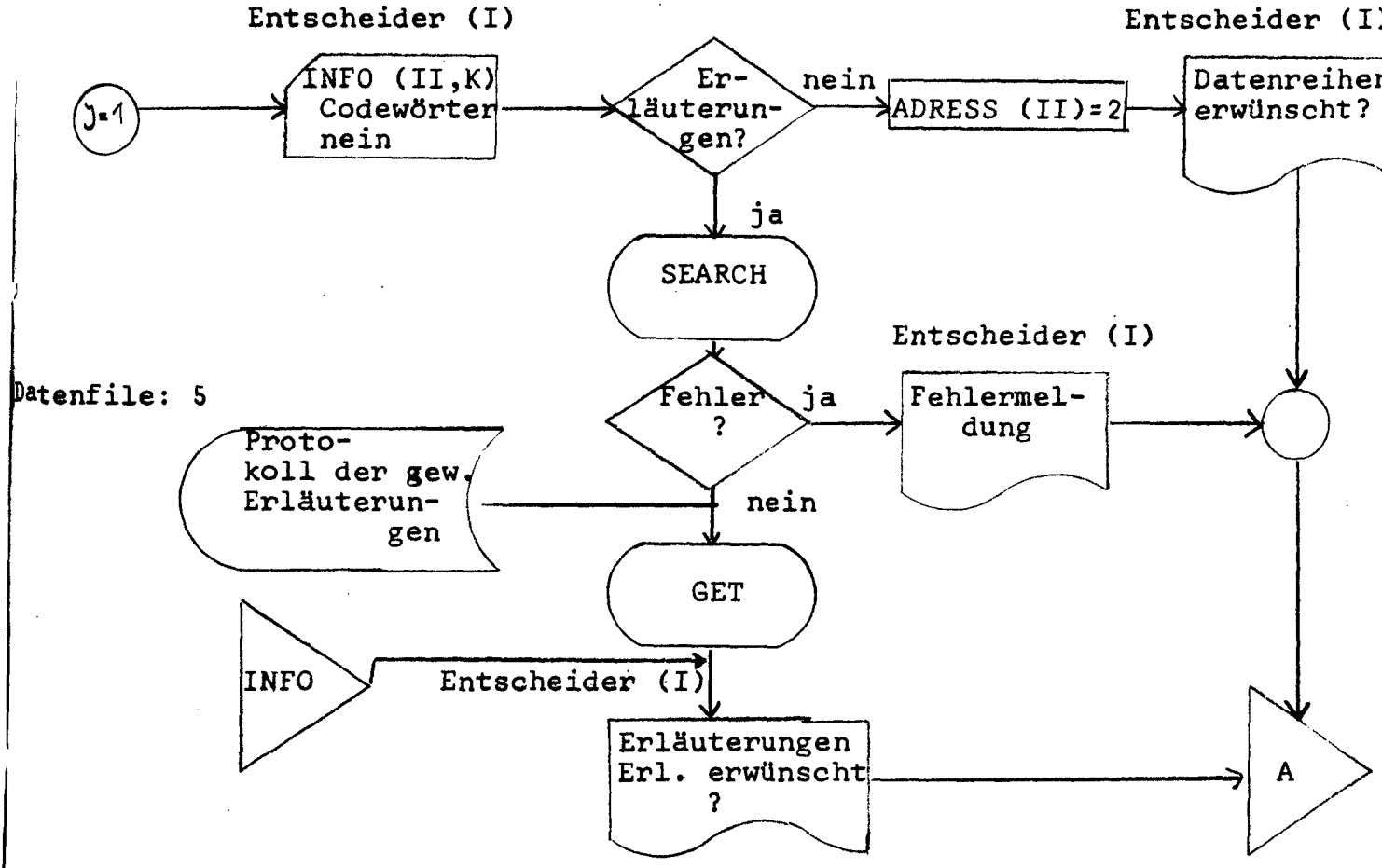


Abb.4 Zweite Adresse: Lesen und Speichern der vom I-ten Entscheider gewünschten Datenreihen in Arrays

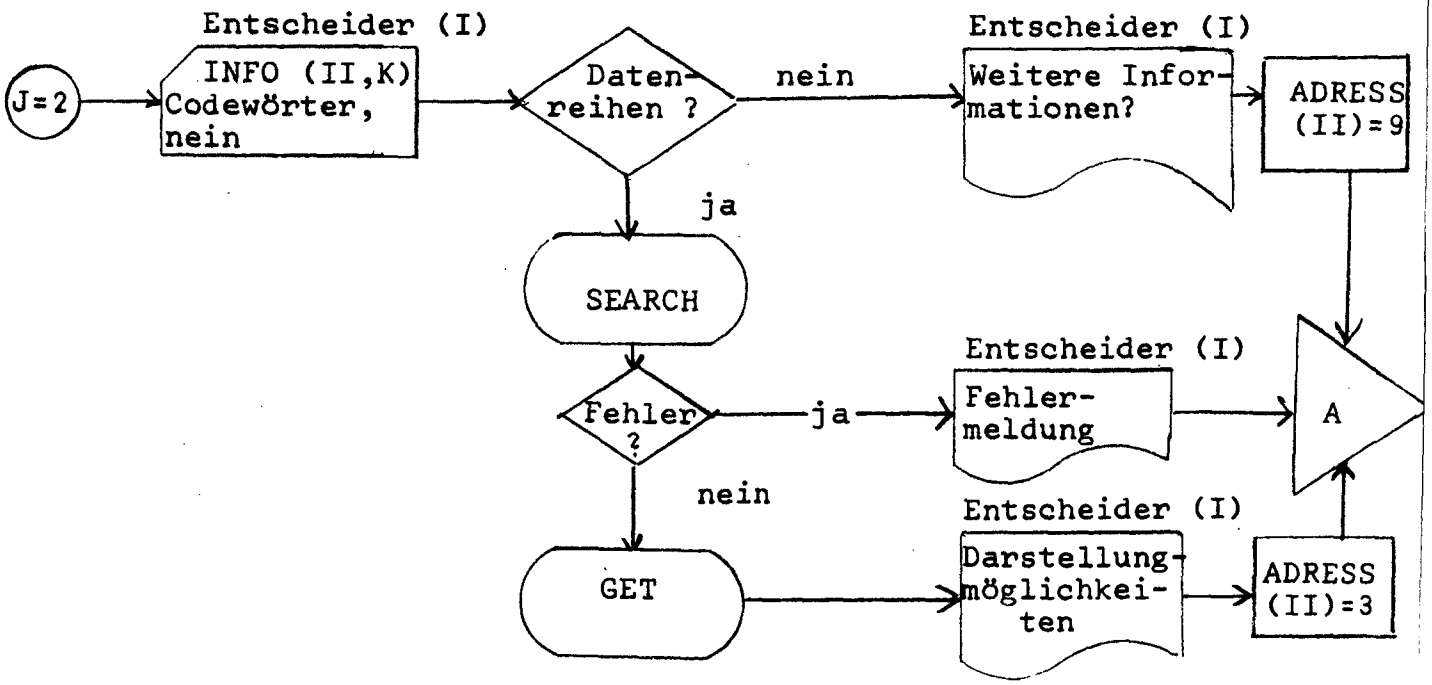


Abb. 12 Eingabe der Antwort auf eine Frage (Prognoseschätzung)

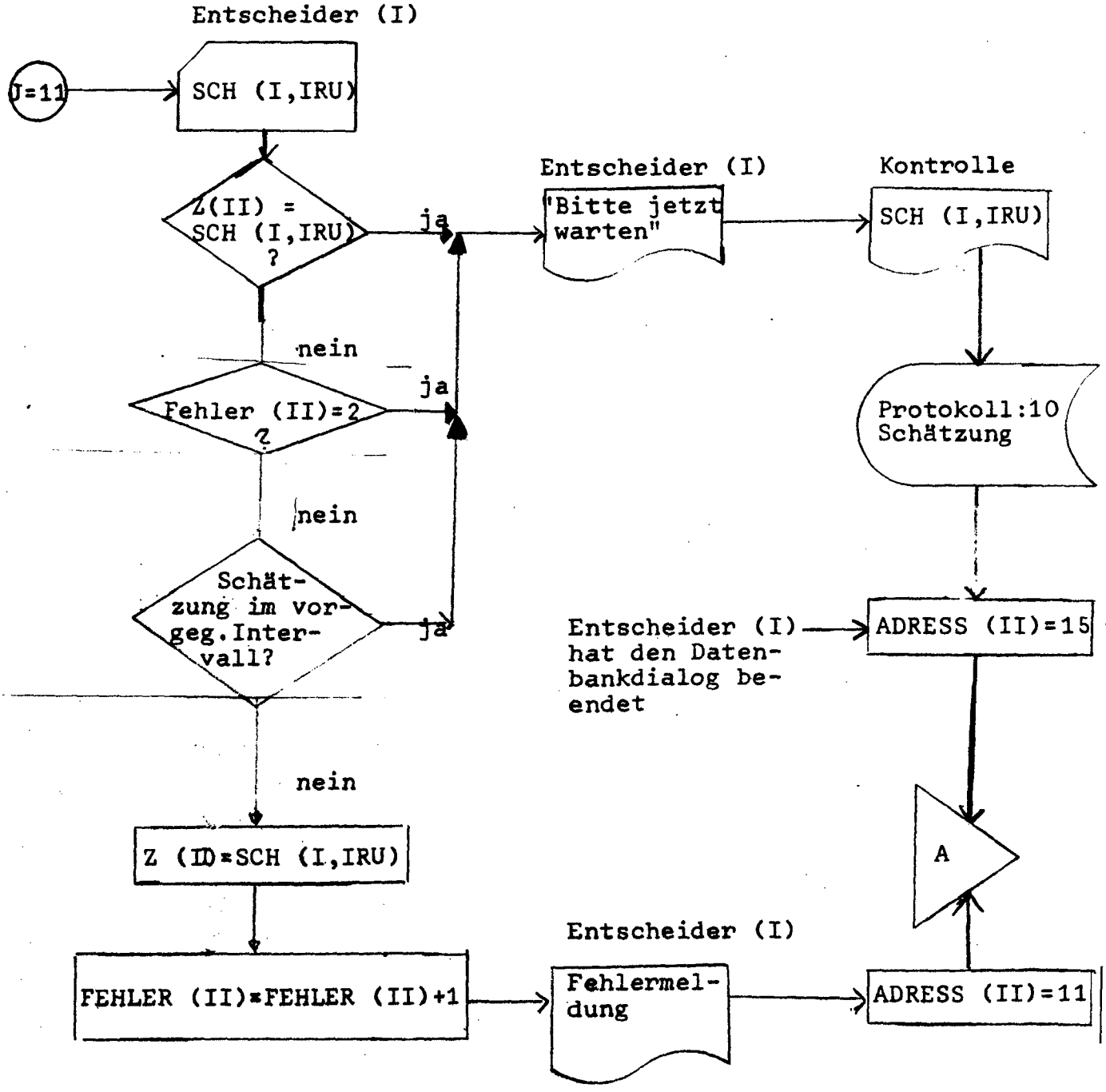
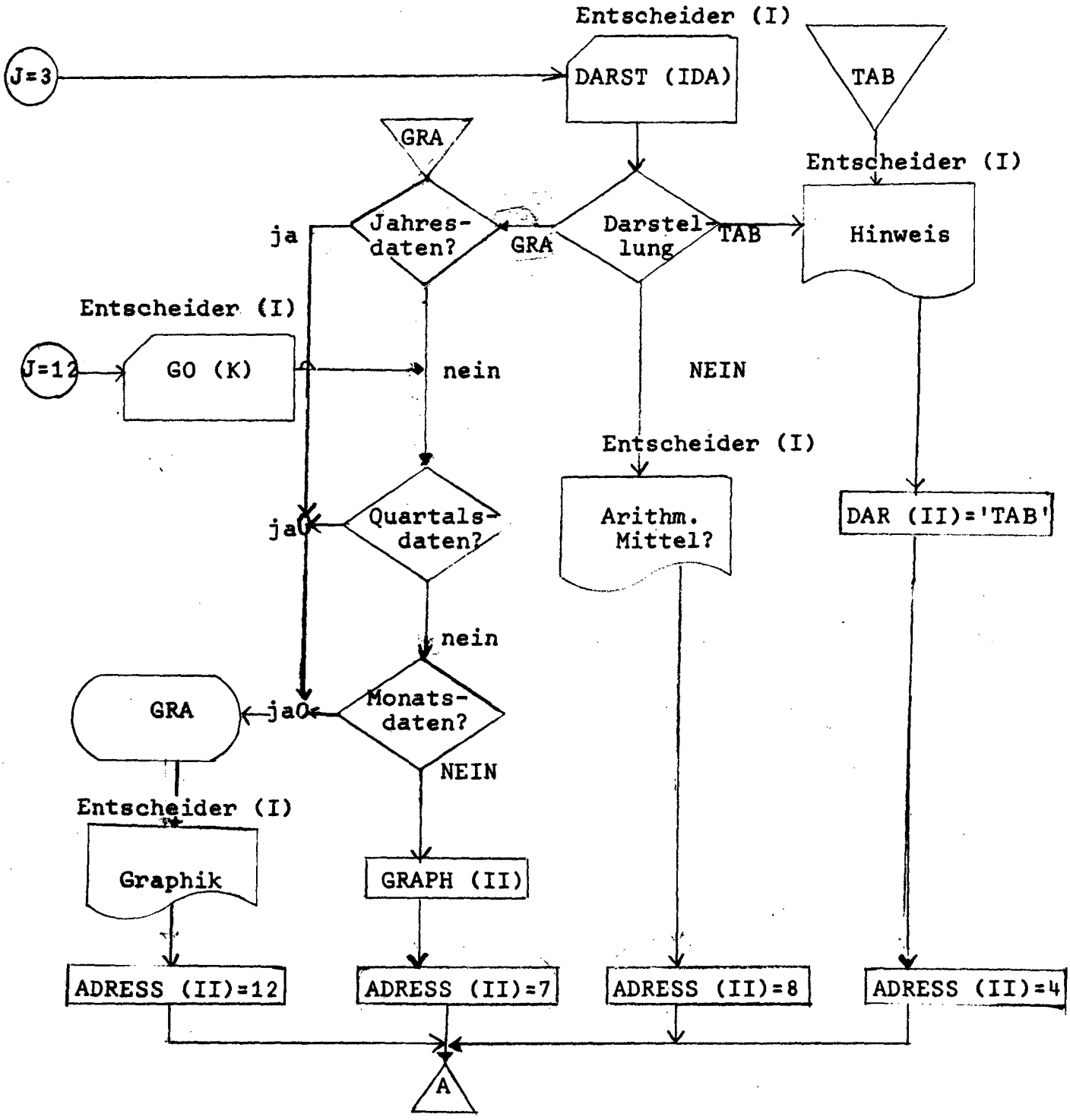
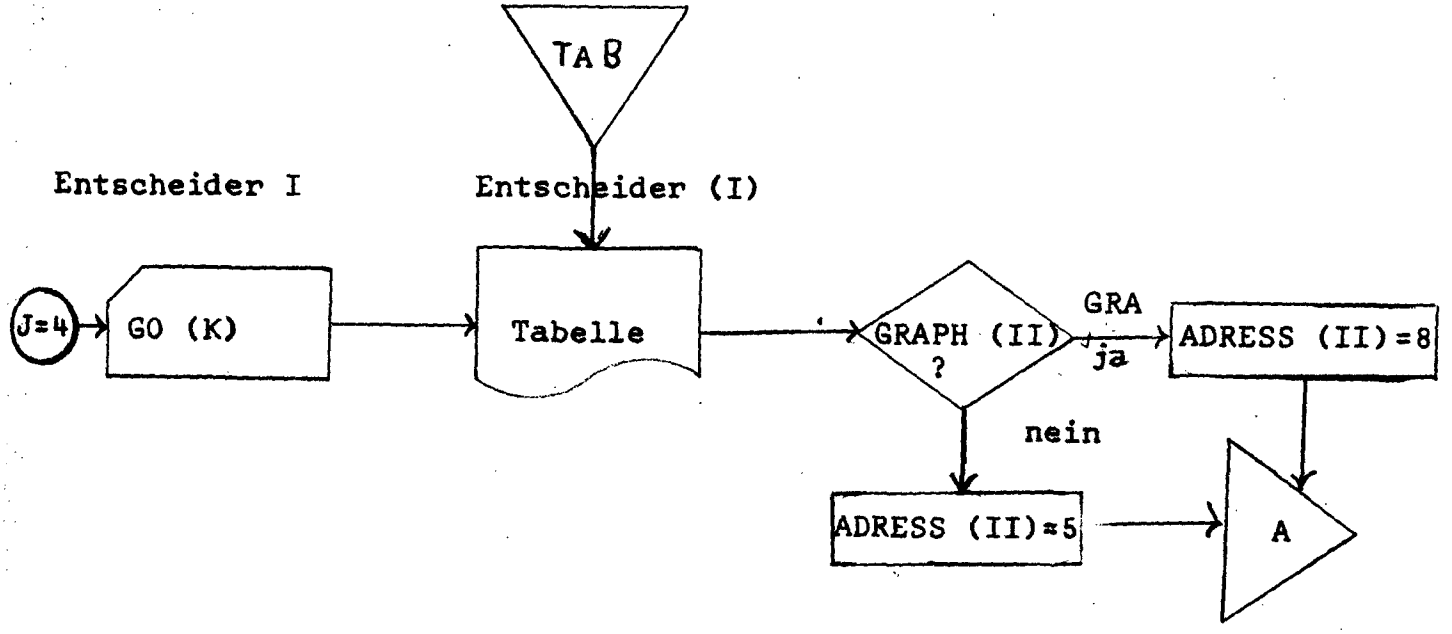


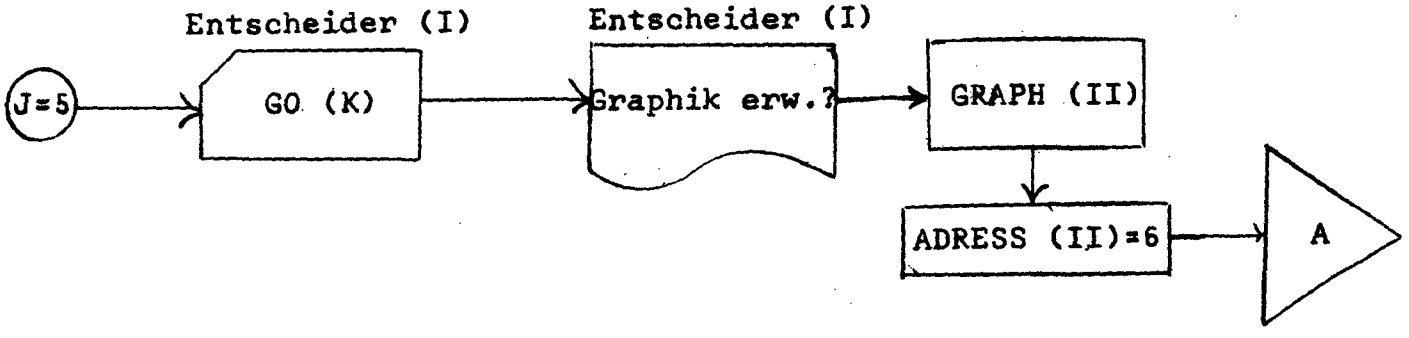
Abb. 5 **Wahl der Darstellungsform:** Tabelle, Graphik oder Arithmetisches Mittel



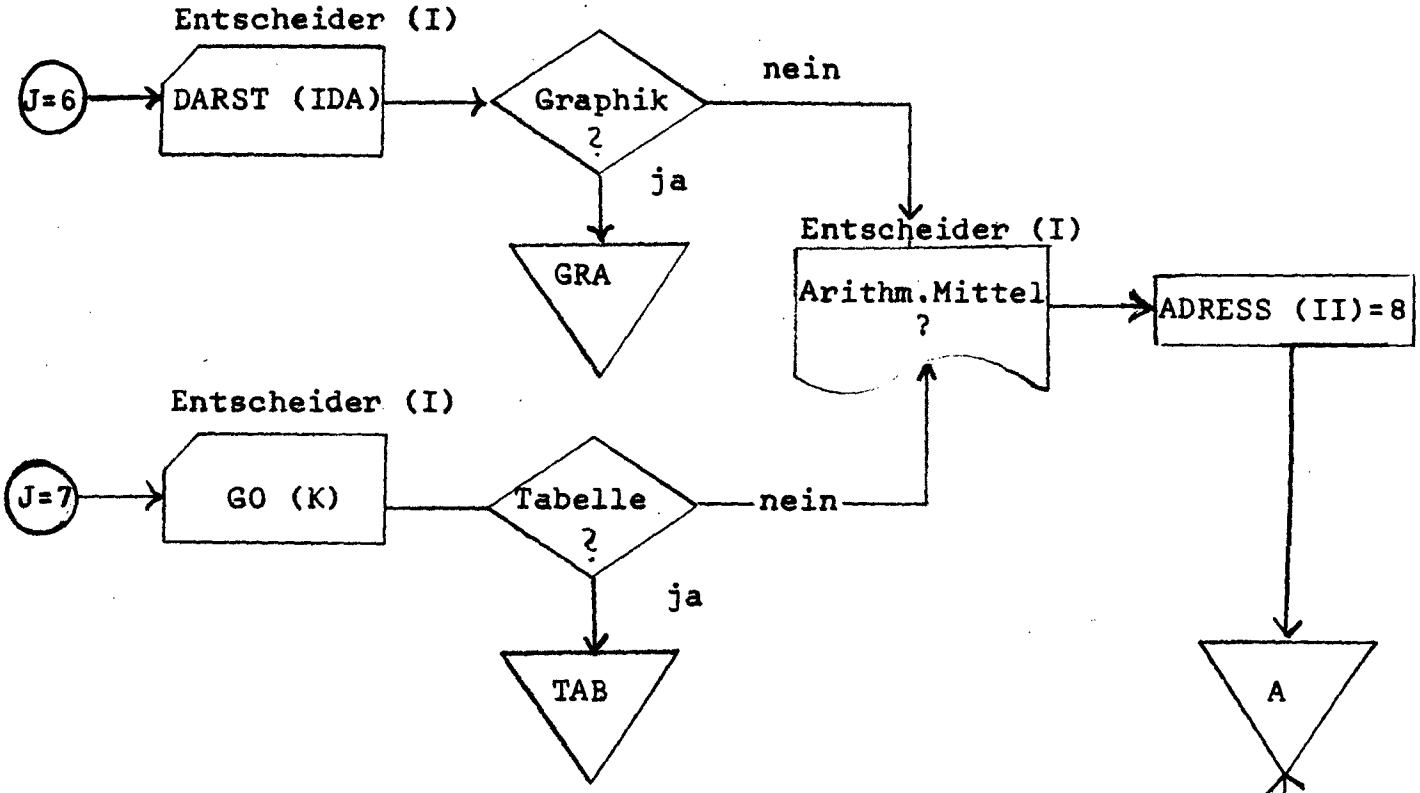
6 Ausgabe der Tabelle



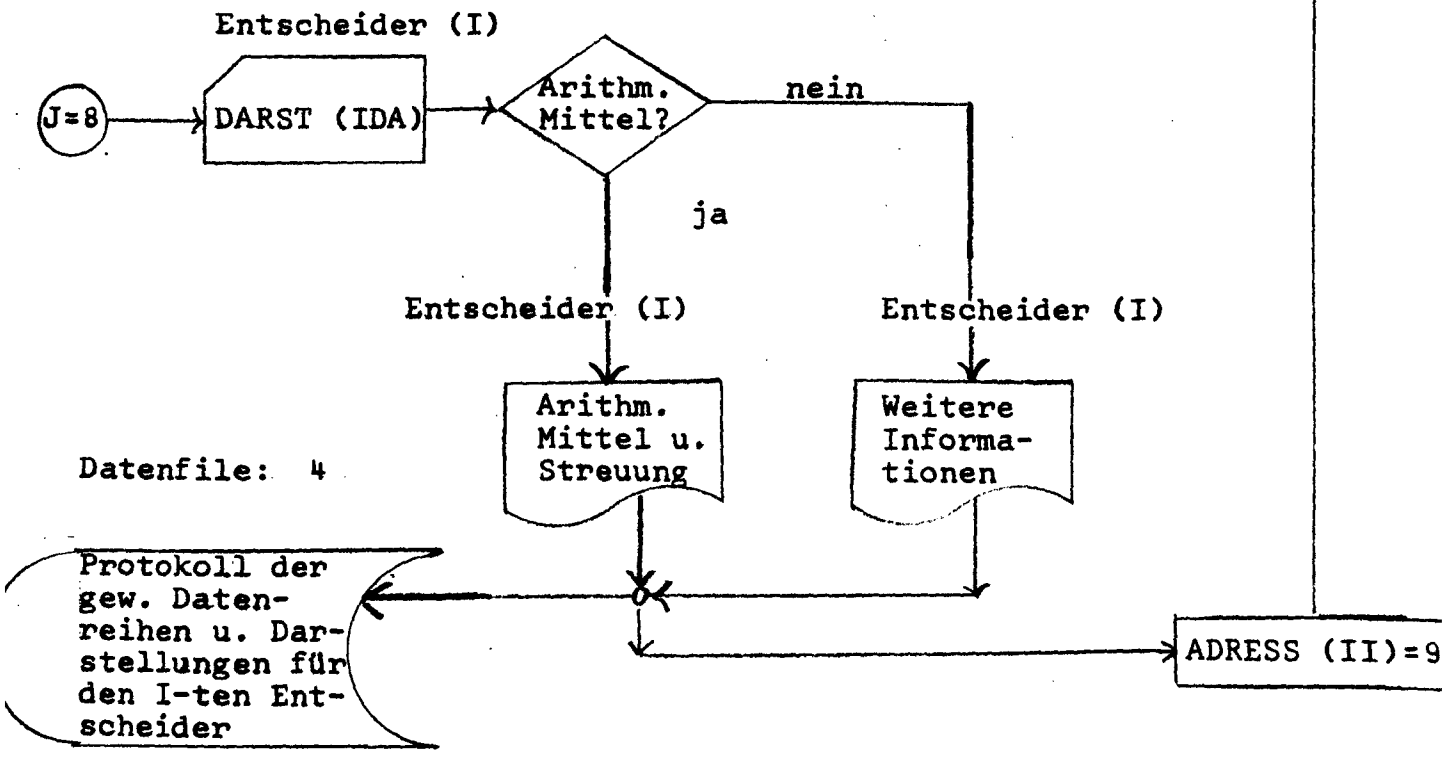
7 Graphische Darstellung erwünscht:



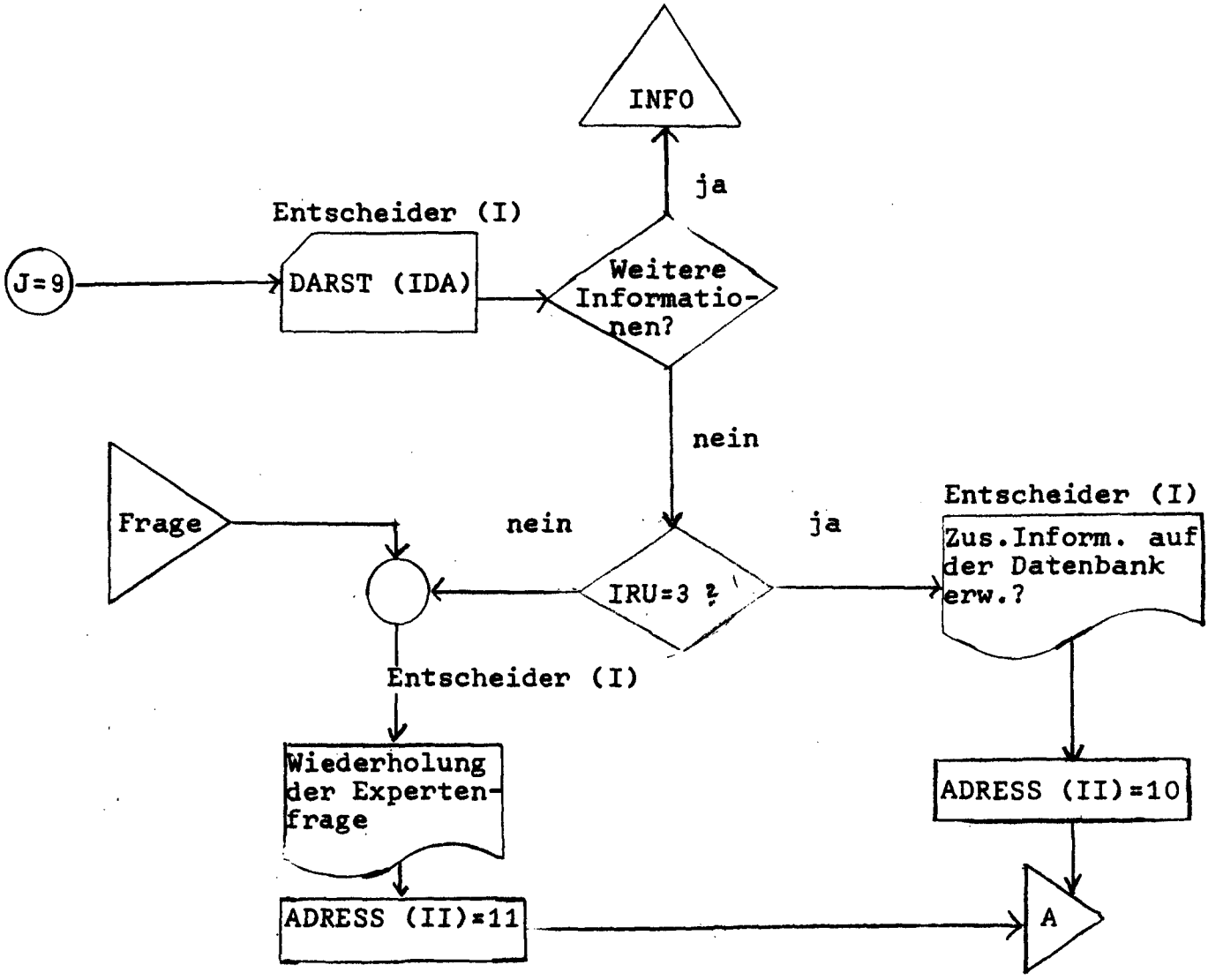
8 Ausgabe der Tabelle bzw. der Graphik



9 Ausgabe des arithmetischen Mittels und der Streuung:



Ab.10 Fortsetzung des Datenbankdialoges



Ab.11 Eingabe von erwünschten Informationsarten, die zusätzlich in die Datenbank aufgenommen werden sollten

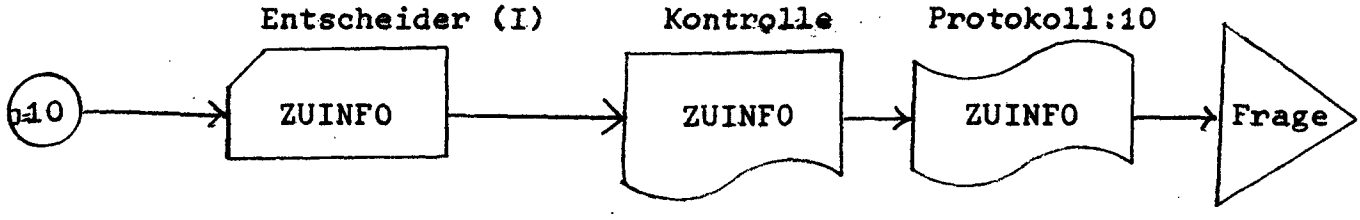
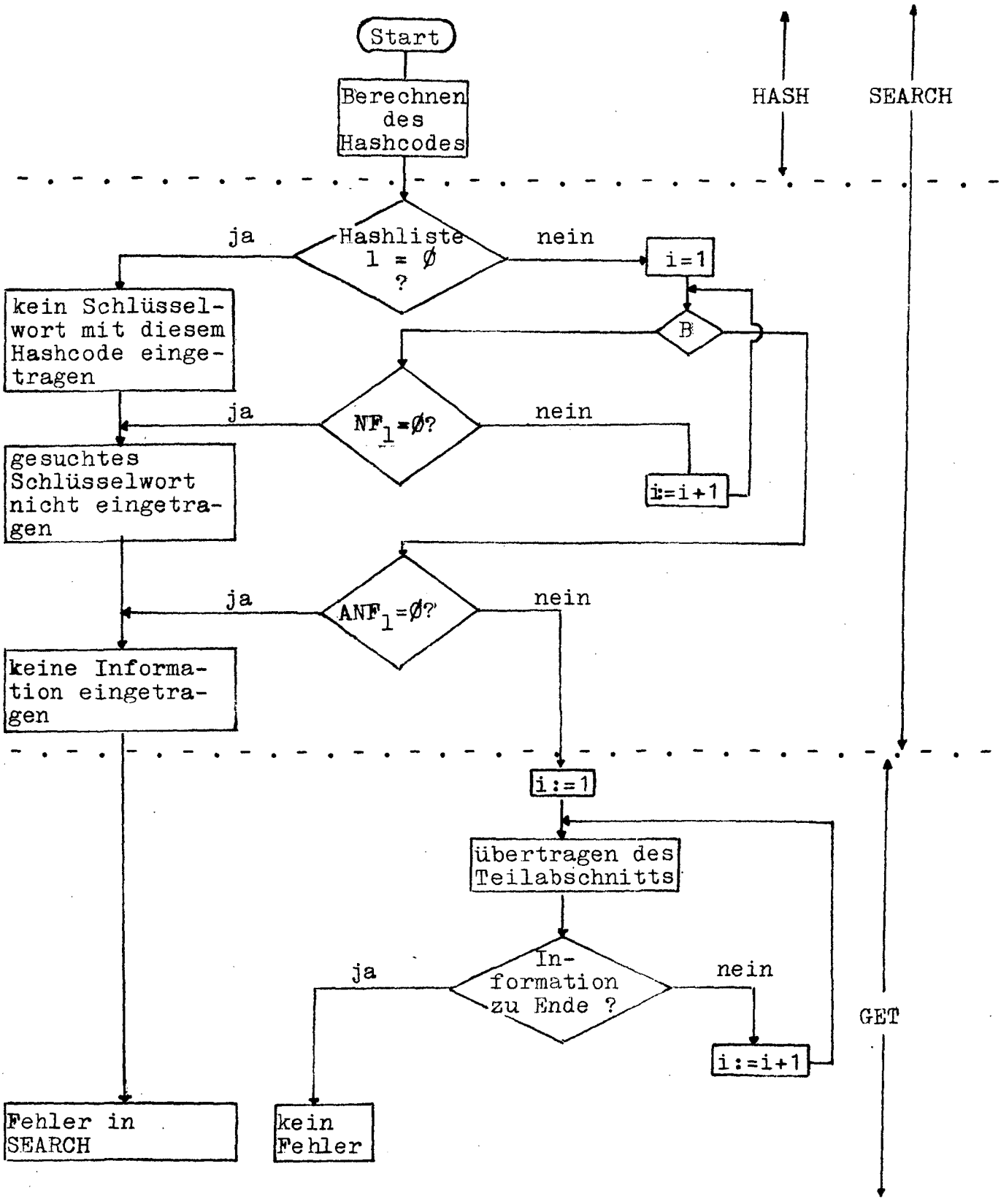
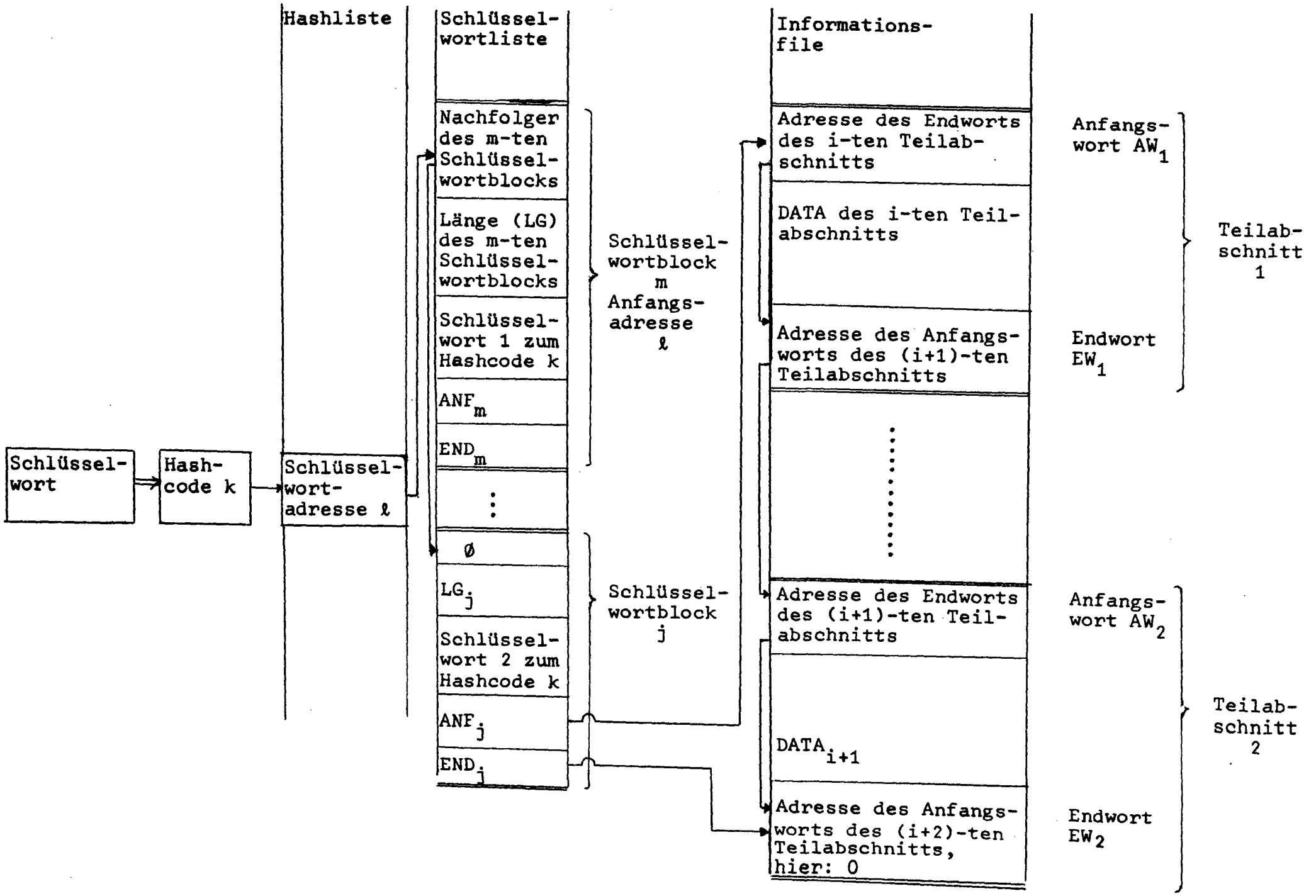


Abb. 12 : s. nach Abb. 4

Abb. 13 : System der Informationsauffindung (SEARCH) und Übertragung (GET)



B : Schlüsselwort ı in der Schlüsselwortliste vorhanden ?


Hashliste
Schlüssel-wortliste
Informations-file
Schlüssel-wort
Hash-code k
Schlüssel-wort-adresse ℓ
Nachfolger des m-ten Schlüssel-wortblocks
Länge (LG) des m-ten Schlüssel-wortblocks
Schlüssel-wort 1 zum Hashcode k
ANF$_m$
END$_m$
Schlüssel-wortblock m
Anfangs-adresse ℓ
Adresse des Endworts des i-ten Teilab-schnitts
DATA des i-ten Teil-abschnitts
Adresse des Anfangs-worts des (i+1)-ten Teilabschnitts
Anfangs-wort AW$_1$
Endwort EW$_1$
Teilab-schnitt 1
Ø
LG$_j$
Schlüssel-wort 2 zum Hashcode k
ANF$_j$
END$_j$
Schlüssel-wortblock j
Adresse des Endworts des (i+1)-ten Teil-abschnitts
DATA$_{i+1}$
Adresse des Anfangs-worts des (i+2)-ten Teilabschnitts, hier: 0
Anfangs-wort AW$_2$
Endwort EW$_2$
Teilab-schnitt 2