

Hinz, Holger; Rickenstorf, Kay

Working Paper — Digitized Version

Ein Programmsystem zum Entscheidungsbaumverfahren

Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 252

Provided in Cooperation with:

Christian-Albrechts-University of Kiel, Institute of Business Administration

Suggested Citation: Hinz, Holger; Rickenstorf, Kay (1990) : Ein Programmsystem zum Entscheidungsbaumverfahren, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 252, Universität Kiel, Institut für Betriebswirtschaftslehre, Kiel

This Version is available at:

<https://hdl.handle.net/10419/175390>

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.

Nr. 252

Ein Programmsystem zum Entscheidungsbaumverfahren

Holger Hinz/Kay Rickenstorff*

September 1990

*** Dr. Holger Hinz, cand. rer. nat. Kay Rickenstorff, Institut für Betriebswirtschaftslehre der Universität Kiel, Forschungsstelle für betriebliche Datenverarbeitung und Kommunikationssysteme**

Direktoren:

**Prof. Dr. Andreas Drexl
Prof. Dr. Reinhart Schmidt**

Ein Programmsystem zum Entscheidungsbaumverfahren

1. Problemstellung

Entscheidungsbaumverfahren zur Lösung sequentieller betriebswirtschaftlicher Aufgabenstellungen haben Tradition.¹ Zur softwaremäßigen Unterstützung dieses Verfahrens existieren bereits eine große Anzahl von Programmpaketen, die sich durch unterschiedliche Schwerpunktbildung auszeichnen.² Um die Vorteile dieser Systeme vereinigen zu können, soll hier der Versuch unternommen werden, prototypisch ein System zu entwickeln, das leicht auf benutzerspezifische Anforderungen weiterentwickelt werden kann.

Im folgenden werden zunächst Implementierungsaspekte beschrieben, bevor anhand eines Beispiels die derzeitige Ausbaustufe des Systems dokumentiert wird. Ein Ausblick auf Erweiterungsmöglichkeiten beschließt die Arbeit.

2. Implementierung

2.1 Technische Aspekte

Das Programm ist vollständig in der Programmiersprache C geschrieben.³ Damit es auch in der Lehre eingesetzt werden kann, wurden umfangreiche Ein- und Ausgaberoutinen geschrieben, die einem Benutzer auf einfache Weise ermöglichen, einen Entscheidungsbaum aufzubauen und die Güte eines Ergebnisses abzuschätzen. Aus diesem Grunde entstand ein Programm von 130KByte Quelltext Länge. Benutzt wurde ein Rechner mit einer EGA-Graphikkarte, die standardgemäß vier Bildschirmseiten zur Verfügung stellt. Das Programm ist lauffähig auf IBM-PC mit DOS 3.3.

Um allgemeine Entscheidungsbäume, d.h. Bäume mit beliebig vielen Perioden, Alternativen in jeder Periode und zu jedem Umweltzustand, aufbauen zu können, sind die hier implementierten rekursiven Strukturen sinnvoll,⁴ da sie sich leicht den Benutzerbedürfnissen hinsichtlich Erweiterungen der Eingabedaten anpassen lassen.

¹ Vgl. z.B. Mao, J.C.T.: Decision Trees and Sequential Investment Decisions, Cost and Management, 4 (1968), S. 18-23.

² Vgl. z.B. Software for Decision Analysis: A Review of Riskcalc, Arborist and Supertree, OR/MS Today, Vol 12 No. 5, S. 24-29.

³ Es wurde der Lattice-C-Compiler Version 6.02 eingesetzt.

⁴ Die Strukturen sind in einer Pascal-ähnlichen Notation im Anhang A angegeben.

2.2 Programmablauf

Es können Prozeduren für die

- Eingabe,
- Berechnung,
- Ausgabe und
- Modifikation des Modells und der Daten

unterschieden werden. In Abb. 1 ist das Ablaufschema des Programms grafisch dargestellt.

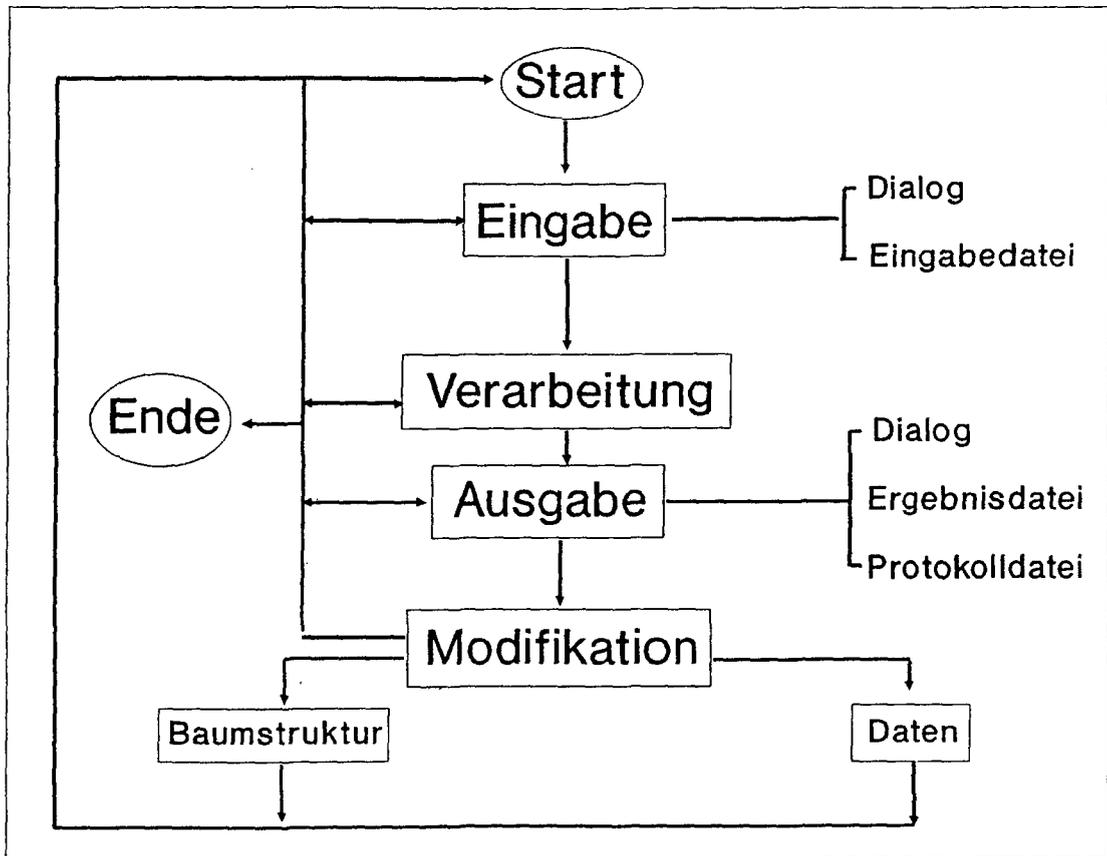


Abbildung 1: Programmablauf

Es besteht die Möglichkeit, die default-Einstellungen der Programmparameter zu verändern. Dazu zählen neben Typ (z.B. Bildschirm, Datei) bzw. Bezeichnungen (z.B. Dateinamen) der Ein- und Ausgabemedien der inhaltliche Umfang der zu erfassenden Modelldaten.

Bei der Eingabe kann zwischen einem Einlesen aus einer Datei⁵ und einer Eingabe über Tastatur und Bildschirm unterschieden werden.

⁵ Vgl. Anhang C

Die Berechnungsprozeduren ('Verarbeitung') bestimmen den maximalen Kapitalwert durch vollständige Enumeration mittels eines rollback-Verfahrens, das auf der Maximierung des Erwartungswertes des Kapitalwertes beruht. Dabei wird also Risikoneutralität des Benutzers unterstellt.

Die Ausgabe kann wahlweise in eine Ausgabedatei, Protokolldatei oder auf den Bildschirm erfolgen. Bei Ausgabe auf den Bildschirm kann zwischen halb- und ganzseitiger Darstellung der Ergebnisse umgeschaltet werden.

Nach der Ausgabe der Ergebnisse besteht die Möglichkeit, das Programm zu verlassen, einen Neustart oder eine Änderung von Daten und Modell durchzuführen. Das Modell kann durch Hinzufügen von Alternativen erweitert und durch Löschen von Alternativen verkleinert werden. Eine neue Verteilung der Umweltzustände führt zu einer Erweiterung durch mehr Umweltzustände, bzw. zu einer Verringerung durch Zusammenfassen mehrerer Umweltzustände zu einem Zustand. Die Modifikation von Daten betrifft die konkrete numerische Ausprägung der Modellstruktur.

Im Anschluß an die Modifikation wird das rollback-Verfahren erneut angewendet und zu dem Punkt 'Berechnen' zurückgekehrt. Der beschriebene Prozeß kann dann von neuem beginnen.

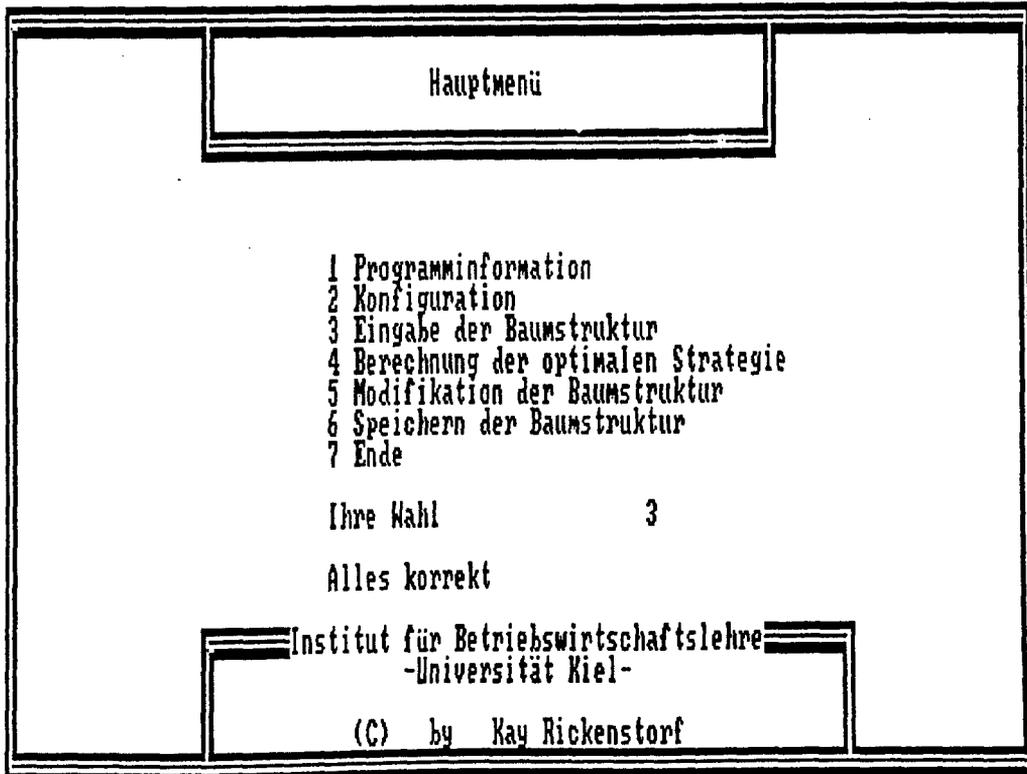


Abbildung 2: Hauptmenü

3. Beispiel

Im folgenden soll das Entscheidungsbaumverfahren für das Problem eingesetzt werden, ob ein großes oder ein kleines Werk in Abhängigkeit von verschiedenen möglichen Umweltzuständen erstellt werden soll.⁶

Nach Programmaufruf wird das Hauptmenü präsentiert (Vgl. Abbildung 2). Hier kann der Benutzer durch Wahl des Menüpunkts 1 dokumentarische Information über Programmablauf und Verwendungszweck erhalten. Durch Wahl des Menüpunkts 'Konfiguration' wird der Benutzer in die Lage versetzt, die Programmparameter zu bestimmen. Insbesondere kann hier die bereits erstellte Eingabedatei 'BLOHM.DAT' als Voreinstellung spezifiziert werden. Diese Baumstruktur wird dann bei 'Eingabe der Baumstruktur' geladen.

Die Berechnung der optimalen Strategie führt zur Ausgabe der Kapitalwerte. Der Bildschirm ist zunächst in zwei Teile gegliedert, in denen die optimale und die nächstbeste Alternative dargestellt werden. An dieser Stelle hat der Benutzer eine Zoom-Funktion zur Verfügung, die eine vergrößerte Darstellung der Baumstruktur - wie in Abbildung 3 - ermöglicht. In der Spalte 'Kapitalwert' sind die Barwerte der Teilstrategien zusammengefaßt.

| Ausgabe der Kapitalwerte | | |
|---|--------------------------------------|--|
| Beste Alternative Kapitalwert [A(1,1)]_kleines_Werk 18.380 | | Nächstbeste Alternative Kapitalwert [A(1,2)]_großes_Werk 16.563 |
| Zeitpunkt | Bezeichnung (Entscheidung, Resultat) | Kapitalwert |
| 0 | [A(1,1)]_kleines_Werk | -13.000000 |
| 1 | [H1:A(2,1)]_kl.-hoch-erw. | -2.181818 |
| 1 | [N1:A(2,4)]_kl.-niedrig-nicht-erw. | 2.727273 |
| 2 | [H2/H1]_kl.-hoch-erw.-hoch | 20.748210 |
| 2 | [N2/H1]_kl.-hoch-erw.-niedrig | 0.576339 |
| 2 | [N2/N1]_kl.-niedrig-nicht-erw.-niedr | 7.780579 |
| 2 | [H2/N1]_kl.-niedrig-nicht-erw.hoch | 1.729018 |

Abbildung 3: Optimale Strategie

⁶ Vgl. Blohm, H.;Lüder,K.: Investition, 6. Aufl., München 1988, S. 253-255.

Im Beispiel wird der Bau eines kleinen Werks als optimale Strategie mit einem Kapitalwert von 18.38 ausgewiesen. Die zugrundeliegende Baumstruktur erstreckt sich hier über zwei Perioden, deren Anfangs- und Endpunkte durch Entscheidungs- bzw. Ergebnisknoten gekennzeichnet sind. Die Bezeichnung dieser Knoten wird im Prinzip dem Benutzer überlassen, wobei es sich als sinnvoll erweist, einen systematischen Bezug zum Problem herzustellen. In Abbildung 3 wurden gleichzeitig zwei Darstellungsformen realisiert. In den geklammerten Ausdrücken findet sich eine eher formalisierte Abbildungstechnik. Die mit A() bezeichneten Elemente stellen Entscheidungsalternativen dar, die jeweils noch durch ein Zweitupel weiter spezifiziert werden. Die erste Komponente steht für die Periode, während die zweite lediglich einen Index zur Durchnummerierung der Alternativen darstellt. Die übrigen Elemente stellen Ergebnisknoten dar, die durch eine Abfolge von Umweltzuständen entstanden sind. Als weitere Möglichkeit der Darstellung wurde exemplarisch der Versuch einer Verbalbeschreibung der Knoten unternommen.

Im Ergebnis wird deutlich, daß das Erreichen des optimalen Kapitalwerts hauptsächlich vom Eintreten der Ereignisfolge [H2/H1] abhängt.

Nach der Optimierung können die oben beschriebenen Modifikationen hinsichtlich der Modellstruktur sowie der Daten durchgeführt werden (Vgl. Abbildung 4).

| Modifikation | |
|------------------|--------------------------------------|
| Modelländerungen | |
| 1 | Bezeichnungen löschen |
| 2 | Bezeichnungen hinzufügen |
| 3 | Verteilung der Umweltzustände ändern |
| Datenänderungen | |
| 4 | Jahresbezeichnung |
| 5 | Wahrscheinlichkeiten |
| 6 | Zinssätze |
| 7 | Zahlungen |
| A | Abbruch |
| Ihre Wahl | 1 |
| Alles korrekt | |

Abbildung 4: Modifikation

4. Ausblick

Der vorgestellte Prototyp zeichnet sich durch flexible Erweiterungsmöglichkeiten aus. Dies gilt zunächst für die Implementierung weiterer Algorithmen aus dem Gebiet des Operations Research, die die Lösung komplexerer Problemstellungen ermöglichen. Dabei kann zunächst an die Implementierung von Mehrfachzielsetzungen gedacht werden. Weiterhin wäre die Berücksichtigung verschiedener Nutzenfunktionen denkbar. Wünschenswert erscheint auch die Simulation der unsicheren Daten. Hier müssen verschiedene Wahrscheinlichkeitsverteilungen zulässig sein. Nicht zufriedenstellend erscheint bisher die Benutzerunterstützung bei der Erstellung der Baumstruktur. Eine komfortable Oberfläche ist daher notwendig, in der der Benutzer mit graphischen Mitteln edieren kann. Dies erscheint allerdings durch den hier gewählten Aufbau der Datenstrukturen besonders leicht möglich.

Anhang A Datenstrukturen

```

record ALTERNATIVE = begin
{ Nachfolgender Knoten }
    tree,
{ Nach Berechnung gesetzte Zeiger auf das beste und zweitbeste
Ergebnis; Typ ist Zeiger auf Knoten }
    maximaler_Knoten, nächst_bester_Knoten
    : ^KNOTEN;
{ Zeiger auf rechten und linken Teilbaum zum Aufbau von
verketteten Listen der Alternativen; rekursive Datenstruktur }
    linker_Baum, rechter_Baum: ^ALTERNATIVE;
{ Zeiger auf den Vorgängerknoten vom Typ Wahrscheinlichkeit }
    vorgänger: ^WAHRSCHEINLICHKEIT_i;
{ Nach Berechnung maximaler Kapitalwert, der zweitbeste
Kapitalwert, an diesem Knoten berechnete Kapitalwert }
    maximaler_Kapitalwert,
    nächstbester_Kapitalwert,
    Alternative_Kapitalwert: FLOAT;
{ Anfangsjahr, Planungsjahr, Anzahl der weiteren Perioden }
    anfangsjahr, planungsjahr,
    endperiode: INTEGER
    end

record WAHRSCHEINLICHKEIT_i = begin
{ Zeiger auf Alternative }
    alternativen: ^ALTERNATIVE;
{ Zeiger auf die Werte der Vorperiode }
    vorherige_Periode: ^KNOTEN;
{ Letzte Wahrscheinlichkeit und nächste Wahrscheinlichkeit }
    wahrscheinlichkeit_i-1,
    wahrscheinlichkeit_i+1: ^WAHRSCHEINLICHKEIT_i;
{ Kapitalwert und Wahrscheinlichkeit an diesem Knoten }
    kapitalwert, wahrscheinlichkeit: FLOAT
    end

record KNOTEN = begin
{ Zeiger auf die Werte für die folgende Periode }
    nächste_Periode: ^WAHRSCHEINLICHKEIT_i;
{ Zeiger auf die Alternative für diese Periode }
    zugehörige_Alternative: ^ALTERNATIVE;
{ Zeiger auf die Eingabewerte für Zinsen, Ein-, Auszahlung }
    Eingabewerte: ^WERTE;
{ Laufzeit in Jahre und Anfangsjahr, Name des Pfades zu diesem
Knoten, aufsummierter Kapitalwert der nachfolgenden Perioden und
der Beitrag dieses Knotens zum Kapitalwert }
    Laufzeit, Anfangsjahr: INTEGER;
    Name: STRING;
    Summe_Kapitalwert, Kapitalwert: FLOAT
    end

record WERTE = begin
{ Werte für Zinsen, Einzahlung, Auszahlung }
    Zins, Einzahlung, Auszahlung: FLOAT;
{ Zeiger auf Werte für das nachfolgende Jahr }
    nächstes_Jahr: ^WERTE
    end

```

Anhang B

Berechnungsprozeduren in einer Pascal-ähnlichen Notation

```

{ Bestimmen des maximalen Kapitalwertes aller Alternativen, der
maximale Kapitalwert wird als Ergebnis zurückgeben; Parameter tree
ist ein Zeiger auf Alternative, aufoderabzins legt fest, ob die
Werte auf- oder abgezinst werden sollen }
procedure berechne (tree: ^ALTERNATIVE; var kapwert: FLOAT;
                   aufoderabzins: BOOLEAN)

{ Zeiger vom Typ Alternative, um rekursiv durch den aufgebauten
Baum das optimale Ergebnis zu bestimmen }
var
  teilbaum: ^ALTERNATIVE;
{ Zeiger vom Typ Knoten, um den optimalen und den nächstbesten Weg
im Baum zu berechnen }
  nodel, node2, node3: ^KNOTEN;
{ Kapitalwert berechnen }
  kapital1, kapital2, kapital3: FLOAT;

begin
{ Vorbesetzen der Variablen }
  teilbaum := tree;
{ größte negative Zahl }
  kapital1 := -9.9e99;
  kapital2 := kapital1;
{ Bester Knoten erster in der Liste }
  ^nodel := ^teilbaum.tree;
{ Kein zweitbestes Ergebnis }
  node2 := NIL;
{ Solange bis der Teilbaum keinen Zeiger auf NIL hat }
  while teilbaum <> NIL do begin
{ Unterroutine, um Strukturen vom Typ Knoten zu berechnen }
  berechneknoten (^teilbaum.tree, aufoderabzins);
  node3 := ^teilbaum.tree;
  kapital3 := ^node3.kapitalwert;
{ Ist der gerade berechnete Kapitalwert > als der beste bisher
gespeicherte }
  if kapital1 < kapital3 then begin
{Ja, demnach wird der zweitbeste Kapitalwert der bisher beste
Kapitalwert, Knoten werden ebenfalls getauscht }
    kapital2 := kapital1;
    kapital1 := kapital3;
    node2 := nodel;
    nodel := node3;
  end {end if }
{ Nein; ist der gerade berechnete Kapitalwert > als der bisher
gespeicherte zweitbeste Kapitalwert }
  else if kapital2 < kapital3 then begin
{ Ja, speichern dieses Wertes in kapital2, festhalten des Pfades
durch Zeiger auf diesen Knoten }
    kapital2 := kapital3;
    node2 := node3;
  end; {end if }
{ gerade berechneter Kapitalwert wird der Alternative zugewiesen }
  ^teilbaum.Alternative_Kapitalwert := kapital3;
{ nächste Alternative in der Liste }
  teilbaum := ^teilbaum.rechter_Baum;
end; { end while }
{ Bester Kapitalwert wird als ergebnis zurückgeliefert }
kapwert := kapital1;

```

```

{ Bester Kapitalwert dieser Alternativenfolge wird in der ersten
Alternative gespeichert }
  ^tree.maximaler_Kapitalwert := kapital1;
{ Bester Pfad wird durch Speichern des Zeigers auf den Knoten mit
dem besten Kapitalwert gespeichert }
  ^tree.maximaler_Knoten := node1;
{ Zweitbesten Kapitalwert dieser Alternativenfolge wird in der
ersten Alternative gespeichert }
  ^tree.nächstbester_Kapitalwert := kapital2;
{ Zweitbesten Pfad wird durch Speichern des Zeigers auf den Knoten
mit dem besten Kapitalwert gespeichert }
  ^tree.nächstbester_Knoten := node2;
end; {en procedure }

{ Kapitalwerte mit der Wahrscheinlichkeit eines Umweltzustandes
multiplizieren; Kapitalwert zurückgeben }
procedure berechneast (tree: ^WAHRSCHEINLICHKEIT_i;
var kapital: FLOAT;
aufoderabzins: BOOLEAN)

var
    rbaum: ^WAHRSCHEINLICHKEIT_i;
    kapital: FLOAT;

begin
{ Initialisieren der Variable }
    rbaum := tree;
    kapital := 0.0;
{ Solange der Zeiger auf Wahrscheinlichkeit_i nicht NIL ist }
    while rbaum <> NIL do begin
{ Berechnungsprozedur für Datentyp Alternative aufrufen }
        berechne (^rbaum.Alternativen, kapital, aufoderabzins);
{ Kapitalwert ist der gerade berechnete multipliziert mit der
Wahrscheinlichkeit an diesem Knoten }
        kapital := ^rbaum.wahrscheinlichkeit * kapital;
{ Summierter Kapitalwert ist der Kapitalwert der vorangegangenen
Alternative in dieser Periode }
        kapital := kapital + kapital;
        ^rbaum.kapitalwert := kapital;
{ Nächster Wahrscheinlichkeitswert }
        rbaum := rbaum.Wahrscheinlichkeit_i+1
    end {end while }
end; {end procedure }

```

```

{ Kapitalwert eines Knotens aus den Daten, wie Absatz, Preis,
Habenzins bestimmen und als Ergebnis die maximalen Ergebnisse der
nachfolgenden Perioden dazu addieren und als Ergebnis
zurückliefern. }
procedure berechneknoten (var node: ^KNOTEN; aufoderabzins:
BOOLEAN)
var
    hilfsPointer: WAHRSCHEINLICHKEIT_i;
    maxkapwert: FLOAT;

begin
{ Initialisieren der Variable }
    hilfsPointer := ^node.nachfolger;
    ^node.Kapitalwert := 0.0;
{ Kapitalwert bestimmen aus Zins, Ein- und Auszahlung }
    berechnewerte (^node.Eingabewerte, ^node.Kapitalwert,
        ^node.Anfangsjahr, aufoderabzins);
{ Wenn hilfsPointer nicht NIL ist => eine weitere Periode ist zu
berechnen }
    if hilfsPointer <> NIL then begin
        berechneast (hilfsPointer, maxkapwert, aufoderabzins);
        ^node.Summe_Kapitalwert := ^node.Kapitalwert + maxkapwert;
    end
    else ^node.Summe_Kapitalwert := ^node.Kapitalwert;
end; {end procedure }

{ Kapitalwert aus
Preis * Absatz)/(1 + Habenzins) - Auszahlung / (1 + Sollzins)
bestimmen
}

procedure berechnewerte (var eingaben: WERTE; var kapwert: FLOAT;
    anfangsjahr: INTEGER; aufoderabzins: BOOLEAN)

var
    hilfsPointer: WERTE;
    auszahlung, hilfsauszahlung: FLOAT;
    laufzeit: INTEGER;

begin
{ Variable initialisieren }
    hilfsPointer := eingaben;
    kapwert := 0.0;
    laufzeit := 0;
{ Solange hilfsPointer nicht den Zeigerwert NIL hat }
    while (hilfsPointer != NULL) begin
        auszahlung := ^hilfsPointer.einzahlung -
            ^hilfsPointer.auszahlung;
        kapitalwert (^hilfsPointer.zins, auszahlung, aufoderabzins,
            laufzeit + anfangsjahr, auszahlung);
        auszahlung := auszahlung - hilfsauszahlung;
        kapwert = kapwert + auszahlung;
        laufzeit := laufzeit + 1;
    { nächste Eingabewerte verarbeiten }
        hilfsPointer := ^hilfsPointer.nächstes_Jahr
    end {end while }
end; {end procedure }

```

```

{ Funktion zur Bestimmung von Kapitalwerten,
  ergebnis = kapital * faktor; faktor wird in dieser Prozedur
  bestimmt durch die Angabe der Zinsen und der Perioden
  (Jahreszahlangabe) }
procedure kapitalwert (zins, kapital: FLOAT;
                      aufzinsungoderabzinsung: BOOLEAN;
                      anzahlperioden: INTEGER; var ergebnis: FLOAT)

var
    faktor, zwischenergebnis: FLOAT;
    zaehler: INTEGER;

begin
    if anzahlperioden = 0 then ergebnis := kapital;
    else begin
    { Berechnung des Faktors }
        faktor := 1 + zins / 100;
        zwischenergebnis := faktor;
        for zaehler := anzahlperioden to 1 step -1
            zwischenergebnis := zwischenergebnis * faktor;
        if aufzinsungoderabzinsung then
            faktor := 1 / zwischenergebnis;
        else faktor := zwischenergebnis;
    { Ende der Berechnung für den Faktor }
        ergebnis := kapital * faktor;
    end {end for }
end; {end procedure }

```

Anhang C

Aufbau einer Eingabedatei zur Spezifikation der Baumstruktur

Allgemeines

Attributbezeichnungen werden kleingeschrieben, Werte werden großgeschrieben. Der Punkt ('.') stellt die Konkatenation dar. Die verwendeten Attribute und deren Bedeutung sind:

| Attributbezeichnung | Bedeutung |
|---------------------|--|
| vl | Werteliste |
| v | Wert |
| t | Anzahl der Alternativen |
| pe | Periode |
| p | Wahrscheinlichkeit |
| d | Dauer einer Periode [in Zeiteinheiten] |

Es wird unterstellt, daß die Attribute beim Start leer sind.

Produktionsregeln und Ausprägungen der Attribute für die verschiedenen Programmmzustände

Die generelle Vorgehensweise soll anhand einer beispielhaften Interpretation des folgenden Abschnitts dargestellt werden:

Durch die Produktion 'read_alternative' wird die Anzahl der Alternativen eingelesen und dem Attribut 'v₂' zugewiesen. Die Produktion 'do_alternative' liest entsprechend viele Zweige des Baumes ($V_3 := V_2$). Der Wert der Produktion 'start' wird durch die Konkatenation des Wertes v₂ mit der Werteliste vl₃ bestimmt. Die übrigen Attribute (t₁ pe₁ t₃ pe₃) werden später gesetzt bzw. benötigt.

start_{vl1} t₁ pe₁ ::= read_alternative_{v₂} do_alternative_{v₃} vl₃ t₃ pe₃

$VL_1 := V_2 \cdot VL_3$

$V_3 := V_2$

$T_3 := T_1$

$PE_3 := PE_1$

$\text{do_alternative}_{v_1 v_1 t_1 pe_1} ::= \text{do_read_period}_{v_1 t_2 pe_2}$

Falls $V_1 = 1$

$VL_1 := VL_2$

$T_2 := T_1$

$PE_2 := PE_1$

$\text{do_alternative}_{v_1 v_1 t_1 pe_1} ::= \text{do_read_period}_{v_1 t_2 pe_2}$
 $\text{do_alternative}_{v_3 v_3 t_3 pe_3}$

Falls $V_1 > 1$

$VL_1 := VL_2 \cdot VL_3$

$T_2 := T_1$

$T_3 := T_1$

$PE_2 := PE_1$

$PE_3 := PE_1$

$\text{do_read_period}_{v_1 t_1 pe_1} ::= \text{read_name}_{v_2}$ $\text{read_times_of_periods}_{v_3}$
 $\text{read_beginning_year}_{v_4}$
 $\text{read_duration_of_period}_{v_5}$
 $\text{do_read_value}_{v_6 t_6 pe_6 d_6}$

Falls $T_1 = 0$

$VL_1 := V_2 \cdot V_3 \cdot V_4 \cdot V_5 \cdot VL_6$

$T_6 := T_1$

$PE_6 := PE_1$

$D_6 := V_5$

$\text{do_read_period}_{v_1 t_1 pe_1} ::= \text{read_name}_{v_2}$
 $\text{read_duration_of_period}_{v_3}$
 $\text{do_read_value}_{v_4 t_4 pe_4 d_4}$

Falls $T_1 > 0$

$VL_1 := V_2 \cdot V_3 \cdot VL_4$

$T_4 := T_1$

$PE_4 := PE_1$

$D_4 := V_3$

do_read_value_{v11} t1 pe1 d1 ::= read_pay_off_{v2}
 read_lending/borrowing_rate_{v3}
 read_pay_in_{v4}
 do_read_probability_{v15} t5 pe5 p5

Falls $D_1 = 1$

$VL_1 := V_2 \cdot V_3 \cdot V_4 \cdot VL_5$

$T_5 := T_1$

$PE_5 := PE_1$

$P_5 := 0.0$

do_read_value_{v11} t1 pe1 d1 ::= read_pay_off_{v2}
 read_lending/borrowing_rate_{v3}
 read_pay_in_{v4}
 do_read_value_{v15} t5 pe5 d5

Falls $D_1 > 1$

$VL_1 := V_2 \cdot V_3 \cdot V_4 \cdot VL_5$

$T_5 := T_1$

$PE_5 := PE_1$

$D_5 := D_1 - 1$

do_read_probability_{v11} t1 pe1 p1 ::= read_probability_{v2}
 start_{v13} t3 pe3

Falls $P_1 + V_2 = 1.0$ und $PE_1 > 0$

$VL_1 := V_2 \cdot VL_3$

$T_3 := T_1 + 1$

$PE_3 := PE_1$

do_read_probability_{v1} t1 pe1 p1 ::= read_probability_{v2}
 start_{v13} t3 pe3
 do_read_probability_{v14} t4 pe4 p4

Falls $P_1 + V_2 < 1.0$ und $PE_1 > 0$

$VL_1 := V_2 \cdot VL_3 \cdot VL_4$

$T_3 := T_1 + 1$

$T_4 := T_1$

$PE_3 := PE_1 - 1$

$PE_4 := PE_1$

$P_4 := P_1 + V_2$

read_alternative_{v1} ::= INTEGER

V_1 aus N

read_times_of_periods_{v1} ::= INTEGER

V_1 aus N

read_name_{v1} ::= STRING

V_1 aus $\{a, \dots, z, A, \dots, Z\}^*$

read_beginning_year_{v1} ::= INTEGER

V_1 aus N

read_duration_of_period_{v1} ::= INTEGER

V_1 aus N

read_pay_off_{v1} ::= FLOAT

V_1 aus R

read_lending/borrowing_rate_{v1} ::= FLOAT

V_1 aus R

readpay_in_{v1} ::= FLOAT

V_1 aus R

read_probability_{v1} ::= FLOAT

V_1 aus (0, 1]