Grundlagen wissenschaftlichen Arbeitens

(winter semester 2004/05)

Thema:

„The programming language PASCAL“

Autor: Ratko Trajanovski
Matrikelnummer: 0327178
Studienkennzahl: 535

Leiter: Prof.Dr***. Franz Puntigam***

Zur Sprache PASCAL

PASCAL ist eine problemorientierte höhere Programmiersprache. Berücksichtigt man Leistungsfähigkeit, logische Struktur, Einfachheit und die leichte Erlernbarkeit insgesamt, so dürfte **PASCAL** zu den modernsten früher existierenden Programmiersprachen zählen.

Pascal wurde 1968 -1974 von Prof. Niklaus Wirth entwickelt. Das Primäre Ziel der Entwicklung war eine Programmiersprache zur Ausbildung von Studenten. In den 70er Jahren Erwarb sich **PASCAL** einen besonderen Ruf als fortschriftliche Programmiersprache. **PASCAL** zählt mittlerweile zu den am weitesten verbreiteten Programmiersprachen.
Die Hauptgründe für die Popularität von **PASCAL** sind:

- ***Die einfache Struktur der Sprache***
- ***Die Unterstützung der strukturierten Programmierung***
- ***Die problemorientierten Datentypen***
- ***Die Möglichkeit mit PASCAL schnell zu fehlerfreien Programme zu kommen***

Die hauptsächlichen Einsatzgebiete von PASCAL waren damals

- ***Ausbildung***
- ***Technische und kommerzielle Anwenderprogramme z.b. CAD Systeme, Datenbankwendungen etc.***
- ***Systemprogramme wie Compiler, Editoren, Bibliotheksysteme etc.***
- ***Spiele wie Schach, Othello, Kalah, etc.***

PASCAL in der Ausbildung

Die erste Programmiersprache, die der Lernende erlernt, hat einen wesentlichen Einfluß auf seinen späteren Programmierstil.Ist die erste Programmiersprache unstrukturiert und voller inhärent unlogischer Sprachelemente, so wird dies den späteren Stil des Programmiers negativ beeinflussen.
PASCAL wurde von Prog. Wirth mit dem Ziel entwickelt, dem Lernenden eine Programmiersprache anzubieten, die es erlaubt das Programmieren als systematische Disziplin darzustellen. **PASCAL** hat wenige Sprachelemente. Alle Sprachelement können logisch erklärt werden. Syntaxregeln sind immer anwendbar – Ausnahmen wurden bewusst vermieden .Alle diese Eigenschaften lassen **PASCAL** als die Sprache der Wahl für alle Ausbildungszwecke erscheinen: An Schulen, Hochschulen und für die eigene Weiterbildung.
Nach der ersten Veröffentlichung von **PASCAL** im Jahre 1968 wurde an der ETH Zürich eine erste versuchsweise Implementierung durchgeführt. Es folgten eine endgültige Definition und Implementierung von **PASCAL**, und nach einigen Jahren des Piloteinsatz wurde 1974 eine revidierte Fassung von **PASCAL** veröffentlicht. Besonders hervorzuheben ist die Tatsache, dass **PASCAL** nicht von einem so genannten Expertenkomitee entwickelt wurde. Stattdessen wurde **PASCAL** aufbauend auf der langjährigen Erfahrung von Prof. Wirth mit verschiedenen Programmiersprachen und Compilern entwickelt.

Lexikalische Elemente

Ein Programm ist eine Folge von Zeichen die nach bestimmten Regeln aneinandergefügt werden müssen. Die elementaren Bestandteilen von Pascal – Programmen sind:

1. ***Spezialsymbole***
2. ***Bezeichner (Namen)***
3. ***Direktiven***
4. ***Zahlen***
5. ***Marken***
6. ***Zeichenketten (Strings)***
7. ***Trenner (Kommentare, Leerzeichen, Zeilenende)***

Diese Bestandteile werden auch lexikalische Elemente genannt. Lexikalische Element sind wiederum zusammengesetzt aus Buchstaben, Ziffern und Sonderzeichen. Groß und kleingeschriebene Buchstaben werden außerhalb von Zeichenketten grundsätzlich nicht unterschieden. Z.b **BEGIN, begin, Begin bzw. INPUT, input, Input bzw. ZEICHEN, zeichen, Zeichen** Werden also jeweils als gleichwertige Schreibweisen behandelt.

1. Spezialsymbolen

Spezialsymbolen bestehen aus den Sonderzeichen (bzw. Zeichenkombinationnen):

+ - * / = < > [ ] . , : ; | ( ) <> <= >= := .. { }

Und den 35 Wortsymbolen (Schlüsselwörter, reservierte bezeichner).

AND, ARRAY, BEGIN, CASE, CONST, DIV, DO, DOWNTO, ELSE, END, FILE, FOR, FUNCTION, GOTO, IF, IN, LABEL, MOD, NIL, NOT, OF, OR, PACKED, PROCEDURE, PROGRAM, RECORD, REPEAT, SET, THEN, TO, TYPE, UNTIL, VAR, WHILE, WITH

Für einige Spezialsymbole gibt es Ersatzdastellungen:

↑ = @ [= (.] = .) and = & not = ¬ or = | ↑ = ^ { = (* }= *)

2. Bezeichner

Bezeichner (Namen, Identifier) werden benutzt zur Berechnung von Konstanten, Typen, Variablen, Prozedure, Funktionen, Parametern, Programmen, Komponenten, Auswahlkomponenten.
Beispiele für Bezeichner, die voneinander verschieden sind;
X Zeit summe readinteger summel AendereDasErsteZeichenEinerZeile AendereDasErsteZeichenEinerSeite
Beispiele ungültiger Bezeichner: ***5 summen, breite + laenge, kein blank***

3 Direktiven

Direktiven sind Weisungen an den Pascal – Compiler. Sie sind vordefinierte Bezeichner die als Ersatz für einen Prozedure oder Funktionsblock verwendet werden. ***Forward*** ist die einzige vorgeschriebene Direktive aller PASCAL Implementierungen. ***External, internal, module, fortran*** sind Beispiele möglicher andere Direktiven. Diese sind jedoch nicht für alle PASCAL Implementierungen definiert.

4. Zahlen

Zahlen in PASCAL sind ganzzahlig (der zugehörige Datentyp ist ***integer)*** oder gebrochen (der Datentyp ist ***Real***). Ganze Zahlen sind Ziffernfolge mit oder ohne Vorzeichen. Der Wert einer Vorzeichen losen ganzen Zahl muss in Intervall 0..Maxint liegen. Maxint ist als größte ganze Zahl der jeweiligen Implementierung definiert.

1 +100 -99 123456	***Integer - Zahlen***

1e10 1E10 -1E10 1E-10 -1E-10 0.1 -0.1 3.141 2.15 123.45678	***Real - Zahlen***

5 Sprungmarken

Als Springmarken werden Ziffernfolgen im Bereich 0..999 verwendet.

6.Zeichenketten

Zeichenketten sind Elemente eines Zeichenvorrats (z.b. EBCDIC), eingeschlossen in Apostrophe.

Beispiele: ***'pascal'; 'Dies ist ein String'; ';' ; 'Der Satz.'***

Groß und Kleinbuchstaben werden hier als unterschiedlich betrachtet. Soll eine Zeichenfolge einen oder mehrere Apostrophe enthalten, mussen diese durch jeweils zwei Apostrophe dargestellt sein

7. Trenner (kommentare,Leerzeichen,Zeilenende)

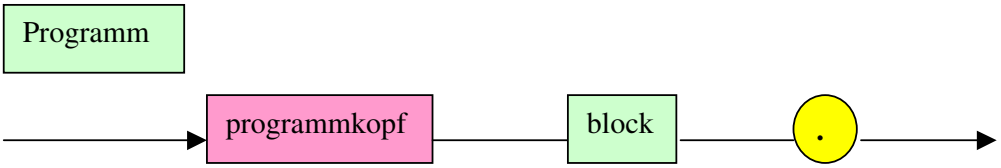
Eine beliebige Folge von Zeichen, eingeschlossen in geschweifte Klammern, gilt als Kommentar.

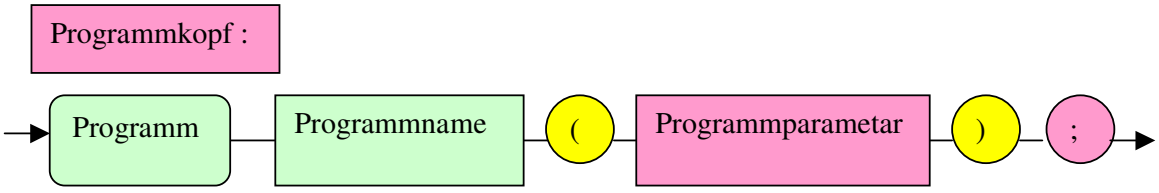
Beispiel:

```
{ Dies ist ein Kommentar}
(*Dies ist auch ein Kommentar *)
{Klammern konnen auch gemischt verwendet werden *)
{Kommentaren koennen ueber
viele Zeilen hinweg fueren}
'{Dies ist kein Kommentar} sondern ein String'
```

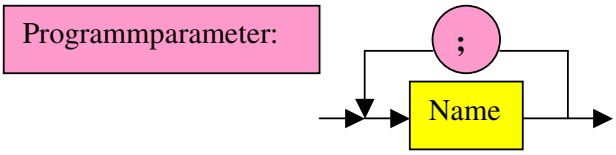
Programmaufbau

Ein Programm hat in PASCAL die folgende Form:



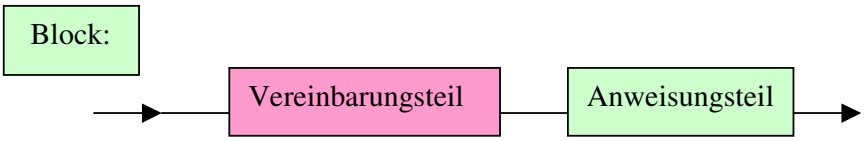


Der Programmname kann beliebig gewählt werden und hat innerhalb des Programms keine weitere Bedeutung. Er kann also auch innerhalb des Programms zur Berechnung anderer Größen benutzt werden. Dieser Name dient vielmehr nur dem Betriebssystem zur Identifizierung des Programms während der Bearbeitung.

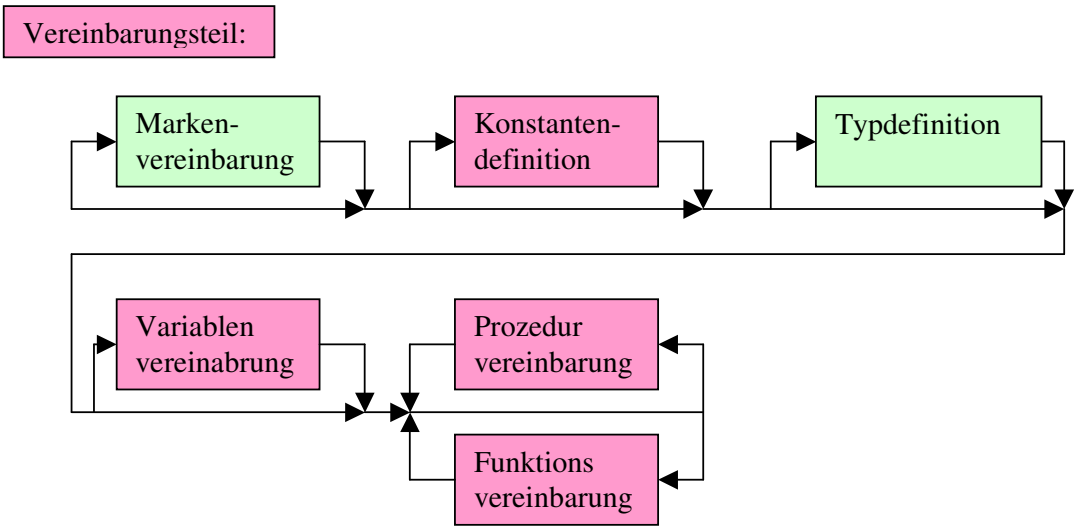


Die Programmparameter bestehen also aus einer Aufzählung von Namen. Es muss sich dabei um die Namen von Files handeln, mit denen es folgende Bewandtnis hat. Größere

Datenmengen werden gewöhnlich in Form von Files (auch Dateien genant) abgespeichert. Dabei kommt es vor, dass diese Files auch außerhalb des Programms existieren sollen, die es also vor und nach dem Lauf eines Programms im Computer geben soll. Es gibt indessen zwei Standardfiles mit Namen ***input*** und ***output*** über die das Programm mit der Peripherie verkehrt.
Das eigentliche Programm wird dann in Form eines Blockes formuliert.

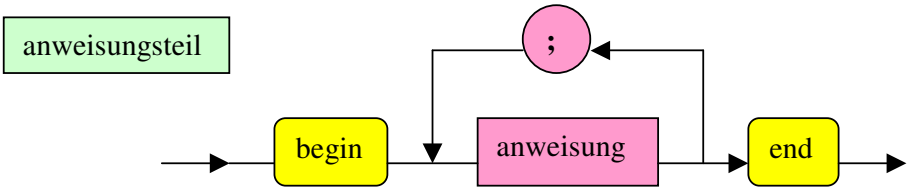


In den Vereinbarungsteil werden die Daten beschrieben, mit denen das Programm etwas machen soll. Der Anweisungsteil enthält dann die Aktionen, die mit den in Vereinbarungsteil charakterisierten Daten durchgeführt werden sollen.



Man beachte den durchgehenden Pfad, wonach jeder Teil und sogar alle fehlen können. Die ersten vier Teile müssen aber, wenn sie vorkommen, eine bestimmte Reihenfolge einhalten. Am Ende des Vereinbarungsteil kommen in beliebiger Reihenfolge die Vereinbarungen der Prozeduren und Funktionen.

Nachdem in Vereinbarungsteil so alle am Programm beteiligten Größen genügend charakterisiert sind, folgt der Anweisungsteil



Im Anweisungsteil werden also die Anweisungen, die den Algorithmus des Problems beschreiben, nacheinander, jeweils durch ein Semikolon getrennt, aufgeführt. Alle Anweisungen werden durch die Wortsymbole ***begin*** und ***end*** eingeschlossen.

Zum Schluss dieses allgemeinen überblick über den Programmaufbau seien einige kleine Programme als anschauungsmaterial angeführt.

```
Programm aufsummieren (input, output):
(* es sollen zahlen eingelesen und aufsummiert  werden, bis ein gewisser betrag erreicht ist.
Es wird unterstellt, dass genügend  zahlen zum einlesen vorhanden sind. *)
Const  betrag = 350.0  (* konstantendefinition*)
Var  summe, x: real      (*variablenvereinbarung*)
Begin
        Summe: = 0;
        While summe <= betrag do
                Begin
                        Read(x):
                        Summe: = summe + x ;

                End;
        Write('die summe betraegt: ', summe);
End.
```

Dazu gehören Daten wie etwa die Zahlen: 13.78 125.5 2.754 198.3 56.56 12.3
Oder noch ein Beispiel dazu

```
Programm Summe (input, output);
Var  i,n, summe: integer;
Begin
        Readln(n);
        Summe: = 0;
                For i: = 1 to n do
                Summe: = summe + i;
        Writeln(summe);
End.
```

Zusammenfasung

PASCAL wurde als Lernsprache konzipiert. Mit dem Ziel Studenten zu lernen saubere Programme zu schreiben. Daher ist die Sprache strenger in der Prüfung und insgesamt sicherer in der Anwendung. Als Lehrsprache wurde leider versäumt eine für die Praxis notwendige Bibliothek an Funktionen, insbesondere für hardwarenahe Operationen und Ein/Ausgabe mitzuliefern.
PASCAL hat echte Defizite wenn es darum geht im kommerziellen Umfeld sehr große Programme zu erstellen, da der ursprüngliche Standard nur eine schlechte I/O Bibliothek enthielt und es an Sprachmitteln fehlt Code zu modularizieren, wie es in andere Sprache wie z.b. in C möglich ist.
Pascal Programme sind sehr gut lesbar. Dazu treten viele dinge bei, die man bei der andere Programmiersprachen weggelassen hat. PASCAL ist leicht zu lernen und ermöglicht die Lösung fast alle Probleme.
Ein Nachteil ist dass PASCAL zu sehr sauberer Programmierung zweigt, was eigentlich kein Nachteil ist aber die Arbeit erschwert.

Literaturliste

- Prof. Dr. ***Rudolf Herschler***, Prof. ***Friedrich Pieper***: **„PASCAL und PASCAL –Systeme**“, Systematische Darstellung für den Anwender. – 4., verb. Aufl. – München; Wien: Oldenburg, 1985.

- ***Kaus Däßler, Manfred Sommer***, **„ PASCAL**“ –Einführung in die Sprache DIN – Norm 66256 Erläuterungen unter Mitarbeit von Albrecht Biedl, zweite Auflage

- *Belli, Fevzi*: **„PASCAL“:** Anleitung zur systematischen Programmierung und Konstruktion zuverlässiger Programme mit Anwendungs Beispielen in Standard und Turbo – Pascal unter Mitarbeit von ***Hausjörg Troebner***
Mannheim; Wien; Zürich: Bi – Wiss – Verl.