

# Wortschatzvarianten von PEARL-Implementierungen

von Prof. Horst Meintzen, Konstanz

## Zusammenfassung

Portabilität beginnt im semantischen Bereich. Zum Vorrat an Schlüsselwörtern von Basic PEARL und von der Ergänzung auf Full PEARL werden Aussagen für 10 Implementierungen über die Anwendbarkeit in einer binären Matrix zusammengestellt. Kurze tabellarische Auswertungen dieser Matrix werden gegeben. Die visuelle Überprüfbarkeit sowie individuelle Auswertungen werden durch eine alphabetische und eine gruppierte Version der Matrix unterstützt. Die verwendete Gruppierung ermöglicht Vergleiche gegenüber anderen Programmiersprachen mit oder ohne derart umfangreiche Möglichkeiten für I/O und für Realzeitbetrieb wie in PEARL bestehend.

Der Wort-Schatz von PEARL ist mehr als ein Schätzchen. Gemäß Definition /1/,/2/ weist er eine stattliche (und begründete) Anzahl von Schlüsselwörtern auf: max. 120 plus 25 Standard-Operatoren. Davon beträgt das Verhältnis im Sprachteil Basic PEARL zu dem in Full PEARL ca. 80 % zu 100 %.

Die gewachsene Entwicklung der Sprache hat ergeben, daß die Grenze zwischen diesen Teilsprachen nicht genau den Unterschied zwischen den mittlerweile existierenden Implementierungen darstellt. Es bestand wohl bereits früher ein Bedarf, die tatsächlich anwendbaren Schlüsselwörter zu den marktgängigen und anderen PEARL-Compilern in einer Zusammenfassung darzustellen.

Eine Liste von Essentials für Programmiersprachen aus Anwender-Sicht umfaßt unter anderem die Stichworte:   Eindeutigkeit, Universalität,  
                  Erlernbarkeit, Praktikabilität,  
                  Verständlichkeit.

Hierzu hat die semantische Ausstattung einer Programmiersprache grundlegende Beiträge zu liefern.  
Statt einer Bewertung des realen Sprachumfanges insgesamt wird im vorliegenden Beitrag eine

vorausgehende (und recht elementar klingende) Zusammenfassung von Aussagen über das Vokabular gegeben. Sie kann damit einige -für Anwender unerfreuliche- Portabilitäts-Probleme einschränken helfen.  
Ein alphabetisches Vokabel-Verzeichnis als Nachschlage-Werkzeug ist sowohl für PEARL-Einsteiger als auch für -Umsteiger erforderlich. Es entstand entsprechend der vorliegenden Form für Studenten der Ingenieur-Informatik und hatte dabei den Schlüsselwörtern in englisch eine Kurzbeschreibung der Bedeutung, bzw.Funktion, in deutsch gegenüberzustellen. Dieses Verzeichnis gestattet es zugleich, den einzuschränkenden und zu verabredenden Sprachumfang anhand des Wortschatzes zu Beginn und am Ende eines Ausbildungsabschnittes übersichtlich zu klären. Die Versuchung, danach den Compiler durch Umdefinieren von Schlüsselwörtern zu testen, hatte im Übungsbetrieb des Verfassers bisher Seltenheitswert. Das Vokabel-Verzeichnis wurde erweitert zu der als T a b e l l e 1 gezeigten Doppeltabelle. Die beiden Einzeltabellen sind nach links bzw. nach rechts von der Schlüsselwort-, d.h.Mittelspalte angeordnet, in welcher sie sich überdecken. Die gemeinsame Spalte enthält in alphabetischer Reihenfolge (Gesamt) die Schlüsselwörter und die Namen der Standard-Operatoren von PEARL. Außer den Langformen von Schlüsselwörtern sind ihre erlaubten Kurzformen, z.B. DCL statt DECLARE, in Klammern daneben angegeben, d.h. nicht in einer eigenen Zeile. In der Mittelspalte links, d.h. rechts vom #-Symbol, wird mit einer binären Angabe ("J" oder "N") verzeichnet, ob das betr. Schlüsselwort bereits in Basic PEARL enthalten ist (oder in der Ergän-

Tabelle 1. Alphabetisches Verzeichnis der Schlüsselwörter von PEARL (Gesamt), von deren Zugehörigkeit zu Basic PEARL und von deren Anwendbarkeit bei den Implementierungen A bis L.  
(siehe nächste Seite)

***** ALPHAB.VERZEICHNIS DER SCHLUESSELWOERTER VON P E A R L (GESAMT) TEIL 1 *****		
IMPLEMENTIERUNGEN	SCHLUESSELWORT	BEDEUTUNG , FUNKTION
:L:K:I:H:G:F:E:D:C:B:A #??		
#####		
+++++ :J ACTIVATE		STARTEN EINER TASK
+++++ :J AFTER		START-BEDINGUNG: ZEITSPANNE
+++++ :J ALL		PERIOD,ZEITDAUER \CONTROL:ALLE FORMATE
+++++ :J ALPHIC		DAT.,INTFAC.-TYP:DRUCKB. ZEICH.(ASCII)
+++++ :J ALT		ALTERNAT. BEI FALL-UNTERSCHIEDG.(CASE)
+++++ :J AT		START-BEDINGUNG: ZEITPUNKT
+++++ :J BASIC		DAT.,INTFAC-TYP:BINAER-WERTE-KOMBIN.
+++++ :J BEGIN		BLOCKBILDUNGS-ANWEISUNG
+++++ :J BIT		DATENTYP: BITKETTE \ SELEKTOR (.BIT)
0++++ :N BOLT		BOLT-VARIABLE
+++++ :J BY		LAUFANW.SCHRITTW. \ I/O-FORMAT-SPEZIF.
+++++ :J CALL		PROZEDUR-AUFRUF
+++++ :J CASE		FALL-UNTERSCHIEDUNG(=SPRUNG-VERTEILER)
+++++ :JJCHARACTER(CHAR)		DATENTYP: ZEICHENKETTEN/SELENT.(.CHAR)
+++++ :J CLOCK		DATENTYP: ZEITPUNKT (=UHRZEIT)
+++++ :J CLOSE		SCHLIESSEN EINER DATION
0++++ :N CONT		DEREFERENZIEREN EIN. REF-OBJEKTES
+++++ :J CONTINUE		FORTSETZEN E.TASK-AKTIVITAET
+++++ :J CONTROL		STEUERGRÖESS.-VEREINBAR. E.DATION
00++++ :N CREATE		KREIEREN EINER DATION
+++++ :J CREATED		BENUTZ.DEFIN. E.DATEI AUF DATION
0++++ :J CYCLIC		FORTSCHALTG.-ATTRIBUT EIN.DATION
+++++ :J DATION		DATENSTATION (-VEREINBARUNG)
+++++ :JJDECLARE (DCL)		DEKLARAT.D.EIGENSCH.V.PEARL-GROESSEN
00++++ :N DELETE		ZERSTOER.V.NICHT MEHR BENOET.DATION
+++++ :J DIM		DIMENSIONIERG.(-VEREINB.)F.DATION
0++++ :J DIRECT		RANDOM-ZUGRIFF AUF ELEM.E.DATION
+++++ :J DISABLE		ABSCHALTUNG EIN.INTERRUPTS
+++++ :JJDuration (DUR)		DATENTYP: DAUER, ZEITSPANNE
+++++ :J DURING		DAUER BIS ENDE(DANACH KEINE WIEDERHOL.)
+++++ :J ELSE		"SONST"
+++++ :J ENABLE		FREIGABE E.INTERRUPTS
+++++ :J END		BLOCKENDE
0++++ :N ENTER		BEGINN SIMULT.ANFORDERG.(BOLT-VARIABLE.)
+++++ :J ENTRY		SPEZIFIK.E.PROZED.\TEIL V.INTFAC-DEKL.
++0++++ :N EVERY		START-BEDING.:PERIOD,ZEITPKT.(VOLLE E.)
000++++ :N EXIT		TEIL V.INTFAC-DEKLARAT.
+++++ :J FIN		ENDE "WENN"-BLOCK, CASE-ABSCHLUSS
+++++ :J FIXED		DATENTYP: GANZZAHL\ DATION-TYP (FILE)
+++++ :J FLOAT		DATENTYP: REELLE ZAHL
+++++ :J FOR		LAUFLISTEN-ANFANG
+++++ :J FORBACK		SEQU.ZUGRIFF, VORW.OD.RUECKW.,A.DATION
+++++ :J FORMAT		DEFINIT. EIN. FORMAT-STEUERLISTE
+++++ :J FORWARD		SEQUENT.ZUGRIFF, VORWAERTS, AUF DATION
0++++ :N FREE		ENDE E.EXKL.BENUTZG.(FREIGABE) BOLT-V.
+++++ :J FROM		GRENZE LAUFBEREICH\QUELLE F.EING.ANW.
+++++ :J GET		EINGABE-ANWEISG. F.ALPHIC-DARSTELLG.
+++++ :J GLOBAL		GLOB. GUELTIGK.VEREINB. ZWISCH.MODULN
+++++ :J GOTO		SPRUNG-ANWEISG.
+++++ :J HRS		ZEITEINHEIT: STUNDE
0++++ :NIDENTICAL (IDENT)		VEREINBARG. PROZEDUR-PARAMETER
+++++ :J IF		BEGINN "WENN"-BLOCK
+++++ :J IN		UEBERTRAGUNGSRICHTG.: EINGABE
0++++ :J INDUCE		EXPLIZ.AUSLOES.E.AUSNAHMEBEDING.(SIGN.)
0++++ :NINITIAL (INIT)		ANFANGSWERT-ZUWEISUNG
0000++++ :N INLINE		DIREKT-EINBETTUNG V.CODE
+++++ :J INOUT		BEIDE UEBERTRAGUNGSRICHTUNGEN
+++++ :JJINTERRUPT(IRPT)		UNTERBRECHUNG (-VEREINB.)
000++++ :N INTFAC		SOFTW.INTFAC (N.-STAND.)ZWISCH.DATIONS
+++++ :J INV		KONSTANT.,INVARIABLE (-DEKLARAT.)
0++++ :N IS		VERGL.OPERAT. (F.REFER.) AUF 'TRUE'
0++++ :N ISNT		VERGL.OPERAT. (F.REFER.) AUF 'FALSE'
0++++ :N LEAVE		ENDE SIMULT.ANFORDERG.(BOLT-VARIABLE.)
+++++ :J LENGTH		LAENGE (-DEKLARAT.)
000++++ :N MATCH		INTFAC-PARAM. F.STEUERKANAL
+++++ :J MAX		MAX.LAENG.ANGABE Z.UEBERTRAG.EINH.TFU
+++++ :J MIN		ZEITEINHEIT: MINUTE
+++++ :J MODEND		MODUL-ENDE
+++++ :J MODULE		MODUL-ANFANG
0++++ :N NIL		LEERE REFERENZ
0++++ :N NOCYCL		GEGENTEIL.FORTSCHALT.ATTRIB.ZU CYCLIC
000++++ :N NOMATCH		GEGENTEIL. INTFAC-PARAM. ZU MATCH
0++++ :J NOSTREAM		GEGENTEIL.FORTSCHALT.ATTRIB.ZU STREAM
#####		
:L:K:I:H:G:F:E:D:C:B:A #	SCHLUESSELWORT	BEDEUTUNG , FUNKTION
-----		

ALPHAB. VERZEICHNIS DER SCHLUESSELWOERTER VON PEARL (GESAMT) TEIL 2		
=====		
:L:K:I:H:G:F:E:D:C:B:A	#	SCHLUESSELWORT BEDEUTUNG , FUNKTION
=====		
:0:++:++:++:++:++:++:++:++	#J	ON REAKT.DEFINIT.AUF SIGNAL(=AUSN.BEDING.)
:0:0:++:0:0:0:0:0:0:++	#N	ONEOF TYP UEBERTRAG.DATEN (=EIN.AUS TYPLISTE
:++:++:++:++:++:++:++:++	#J	OPEN OEFFNEN E.DATION \TEIL E.INTFAC-DEKL.
:0:++:++:0:0:++:0:0:0:0:0	#N	OPERATOR NICHT-STANDARD-OPERATOR (-VEREINBARG.)
:++:++:++:++:++:++:++:++	#J	OUT UEBERTR.RICHT.:AUSG.\FEHL.AUSG.B.CASE
:0:++:++:0:0:++:0:0:0:0:0	#N	PRECEDENCE BENUTZ.DEF.VORRANG(-VEREINB.)F.OPERAT.
:++:++:++:++:++:++:++:++	#J	PRESET ANFANGSWERT-ZUWEISG.F.SEMA-VAR.,-ARRAY
:++:++:++:++:++:++:++:++	#J	PREVENT AUSPLANEN E.TASK-AKTIVITAET
:++:++:++:++:++:++:++:++	#JJ	PRIORITY(PRIO) TASK-PRIORITAET (IN PEARL-MODUL 1)
:++:++:++:++:++:++:++:++	#J	PROBLEM ANFANG PROBLEM-TEIL
:++:++:++:++:++:++:++:++	#JJ	PROCEDURE(PROC) PROZEDUR (-VEREINB.)
:++:++:++:++:++:++:++:++	#J	PUT AUSGABE-ANWEISG. F.ALPHIC-DARSTELLG.
:++:++:++:++:++:++:++:++	#J	READ EINGABE-ANWEISG. F.INTERN-DARSTELLG.
:++:++:0:0:++:++:++:++:++:++	#J	REENT MEHRF. BENUTZBARKEIT E.PROZEDUR
:0:++:++:0:0:++:0:0:0:0:++	#N	REF REFERENZOBJEKT (-VEREINBARG.)
:++:++:++:++:++:++:++:++	#J	RELEASE EXPLIZ.FREIGABE E. SEMA-VARIABLEN
:++:++:++:++:++:++:++:++	#J	REPEAT ANFANG LAUFBEREICH (WIEDERHOLGS.TEIL)
:++:++:++:++:++:++:++:++	#J	REQUEST EXPLIZ.SPERRUNG E.SEMA-VARIABLEN
:0:++:++:0:0:++:0:0:0:0:0	#N	RESERVE EXCLUS.BENUTZG. (=SPERRE) E.BOLT-VAR.
:++:++:0:0:++:++:++:++:++:++	#J	RESIDENT AUSLAGERUNGSFAEHIG,JEDOCH SCHNELLZUGR.
:++:++:++:++:++:++:++:++	#J	RESUME VERZOEG.FREIGABE E.TASK-AKTIVITAET
:++:++:++:++:++:++:++:++	#J	RETURN RUECKSPRUNG-ANWEISG. AUS PROZEDUR
:++:++:++:++:++:++:++:++	#J	RETURNS TYP U.DIMENS. F.ERGBN.AUSG.E.PROZEDUR
:++:++:++:++:++:++:++:++	#J	SEC ZEITEINHEIT: SEKUNDE
:++:++:++:++:++:++:++:++	#J	SEMA SEMAPHOR-VARIABLE
:++:++:++:++:++:++:++:++	#J	SEND AUSGABE-ANWSG. F.BASIC-DARSTELLG.
:0:++:++:++:++:++:++:++:++	#J	SIGNAL REAKTION(-VEREINB.) AUF LAUFZEITFEHLER
:++:++:++:++:++:++:++:++	#JJ	SPECIFY (SPC) SPEZIFIK.V.BEREITS DEKL.PEARL-GROESS.
:0:++:++:++:++:++:++:++:++	#J	STREAM FORTSCHALT.-ATTRIBUT E.DAT.STAT.
:++:++:++:++:++:++:++:++	#J	STRUCT STRUKTUR, VERBUND (-VEREINBARG.)
:++:++:++:++:++:++:++:++	#J	SUSPEND ANHALTEN EIN.TASK-AKTIVITAET
:0:0:++:0:0:0:0:0:0:0:0:0	#N	SYS SYSTEM-REAKTION AUF ON-(SIGN.)ANWSG.
:++:++:++:++:++:++:++:++	#J	SYSTEM ANFANG SYSTEM-TEIL
:++:++:++:++:++:++:++:++	#J	TAKE EINGABE-ANWEISG. F.BASIC-DARSTELLG.
:++:++:++:++:++:++:++:++	#J	TASK TASK (-VEREINBARG.)
:++:++:++:++:++:++:++:++	#J	TERMINATE (VORZEITIGES) BEENDEN E.TASK-AKTIVIT.
:++:++:++:++:++:++:++:++	#J	TFU UEBERTRAGUNGSEINHEIT F.DAT.KANAL
:++:++:++:++:++:++:++:++	#J	THEN "DANN"
:++:++:++:++:++:++:++:++	#J	TO GRENZE LAUFBEREICH\ZIEL F.AUSG.ANWEIS.
:++:++:0:0:++:++:0:0:0:++	#N	TRIGGER AUSLOESEN E.INTERRUPTS
:0:++:++:0:0:++:0:0:0:0:++	#N	TYPE NEUER-TYP -DEFINITION
:++:++:++:++:++:++:++:++	#J	UNTIL ZEITPUNKT F.ENDE V.TASK-WIEDERHOLGN.
:0:0:++:0:0:0:0:0:0:++:++	#N	UPON UNTERE DATION (=GERAET) BEI CREATE
:0:0:++:0:0:0:0:0:0:0:0:0	#N	USING INTFAC-VERWENDG. BEI CREATE E.DATION
:++:++:++:++:++:++:++:++	#J	WHEN TASK-STARTBEDING.: ABHAENG.V.INTERRUPT
:++:++:++:++:++:++:++:++	#J	WHILE "SOLANGE ...GILT"
:++:++:++:++:++:++:++:++	#J	WRITE AUSGABE-ANWEISG. F.INTERN-DARSTELLG.
=====		
:L:K:I:H:G:F:E:D:C:B:A	#	STD.OPERATOR BEDEUTUNG , FUNKTION
=====		
:++:++:++:++:++:++:++:++	#J	ABS ABSOLUT-BETRAG
:++:++:++:++:++:++:++:++	#J	AND LOG. KONJUNKTION (UND)
:++:++:++:++:++:++:++:++	#J	CAT VERKETTUNG (CATENATION) V.CHAR OD. BIT
:++:++:++:++:++:++:++:++	#J	CSHIFT KREISSCHIFT
:++:++:++:++:++:++:++:++	#J	ENTIER GROESSTER GANZZAHLIGER WERT
:++:++:++:++:++:++:++:++	#J	EQ VERGLEICH AUF "GLEICH"
:++:++:++:++:++:++:++:++	#J	EXOR LOG. ANTIVALENZ (EXKLUSIV ODER)
:++:++:++:++:++:++:++:++	#J	FIT LAENGEN-ANPASSUNG, FIXED OD. FLOAT
:++:++:++:++:++:++:++:++	#J	GE VERGLEICH AUF "GROESSER ALS/GLEICH"
:++:++:++:++:++:++:++:++	#J	GT VERGLEICH AUF "GROESSER ALS"
:++:++:++:++:++:++:++:++	#J	LE VERGLEICH AUF "KLEINER ALS/GLEICH"
:++:++:++:++:++:++:++:++	#J	LT VERGLEICH AUF "KLEINER ALS"
:0:++:++:++:++:0:0:++:++:++	#N	LWB UNTERGRENZE V. ARRAY-INDIZES
:++:++:++:++:++:++:++:++	#J	NE VERGLEICH AUF "UNGLEICH"
:++:++:++:++:++:++:++:++	#J	NOT LOG. NEGATION (BITWEISE INVERTIERG.)
:++:++:++:++:++:++:++:++	#J	OR LOG. DISJUNKTION (INCLUSIV ODER)
:0:++:++:0:0:++:0:0:0:0:++	#N	REM REST NACH GANZZAHLIGER DIVISION
:0:++:++:++:++:++:++:++:++	#J	ROUND RUNDUNG
:++:++:++:++:++:++:++:++	#J	SHIFT GESTRECKTER SCHIFT
:++:++:++:++:++:++:++:++	#J	SIGN VORZEICHEN-BESTIMMUNG
:++:++:++:++:++:++:++:++	#J	TOBIT TYPWANDLG.: FIXED IN BIT
:++:++:++:++:++:++:++:++	#J	TOCHAR TYPWANDLG.: FIXED IN CHAR (ASCII)
:++:++:++:++:++:++:++:++	#J	TOFIXED TYPWANDLG.: BIT ODER CHAR IN FIXED
:++:++:++:++:++:++:++:++	#J	TOFLOAT TYPWANDLG.: FIXED IN FLOAT
:++:++:++:++:++:++:++:++	#J	UPB OBERGRENZE V. ARRAY-INDIZES
=====		
:L:K:I:H:G:F:E:D:C:B:A	#	STD.OPERATOR BEDEUTUNG , FUNKTION
=====		

zung auf Full PEARL). In der linken Teiltabelle, und zwar vom #-Symbol ab nach links, sind weitere Spalten, benannt mit "A" bis "L", angeordnet. Diese Spalten repräsentieren mehrere PEARL-Implementierungen, welche von verschiedenen Herstellern stammen, und zwar in unbewerteter Reihenfolge. Die Aussagen in diesen Spalten sind wiederum binär ("+" bzw. "O"), je nachdem, ob das betreffende Schlüsselwort tatsächlich anwendbar ist oder nicht. Sie stellen Angaben dar, die per Umfrage vom Hersteller stammen, bzw. bestätigt wurden. In einigen Fällen wurden Varianten der Anwendbarkeit erkennbar, die mehrwertige statt der binären Aussagen an einigen Plätzen der Matrix erfordern würden oder z.B. Fußnoten zu den binären Aussagen. Dies bedeutet, daß die "+"-Aussagen partiell eingeschränkt werden, also zu optimistisch sind, jedoch z.B. keinen Fehler zur Übersetzungszeit ergeben. Es überschreitet den vorgesehenen Rahmen dieses Beitrages, Details hierzu aufzulisten.

Die in T a b e l l e 1 links gezeigte Einzeltabelle (hier also vereinfacht) stellt eine binäre Matrix dar. Aus der vorliegenden 2-D-Form kann die 3-D-Form als Erweiterung abgeleitet werden. Die Auswertung der vorliegenden Form sei auf folgenden drei Wegen vorgenommen:

1.1 Spalten-Analyse: die Spaltensummen für "+" und für "O" werden getrennt gebildet und ergeben die in T a b e l l e 2 dargestellte Werte-Verteilung.

Diese (Ist-) Werte stehen den Wortschatz-(Soll-) Werten als Anteile gegenüber:

Basic PEARL	≙ Summe "J" = 117
Ergänzg. auf Full PEARL	≙ Summe "N" = <u>28</u>
Full PEARL	145

Bei der Mehrzahl der betrachteten Implementierungen scheint also ein größerer Wortschatz als der zu Basic PEARL installiert zu sein. 100 % von Full PEARL ist nicht vertreten. Es ist zunächst noch nicht erwiesen, daß sich die "+" -Summen auf Basic PEARL konzentrieren.

1.2 Zeilen-Analyse: die Binärkombinationen sämtlicher Implementationen werden zeilenweise gebildet, verschiedenen Klassen zugeordnet und mit den "J"- bzw. "N"- Aussagen über die Zugehörigkeit des Schlüsselwortes zu Basic PEARL verknüpft. Es ergeben sich die in T a b e l l e 3 dargestellten Fälle und Werte-Verteilungen. Bei 42 % der Schlüsselwörter ist entsprechend den Klassen Alpha

plus Gamma die Portabilität vom Wortschatz her nicht eingeschränkt, während bei 58 % entsprechend Beta eingeschränkte Portabilität angezeigt ist. Alpha-Vokabeln sind z.B. FIXED, FLOAT, ACTIVATE; die einzige Gamma-Vokabel ist INLINE.

1.3 Die Spalten-Analyse gemäß 1.1 als Gesamt-Summierung kann verfeinert werden, indem die "+"- bzw. "O"- Aussage je Implementierungs-Spalte mit der "J"- oder "N"-Aussage über die Basic PEARL-Zugehörigkeit verknüpft wird. Da eine visuelle Kontrolle hierzu anhand von T a b e l l e 1 jedoch durch Unübersichtlichkeit behindert wird, sei diese Analyse statt dessen an T a b e l l e 4 orientiert.

Diese Doppeltabelle kommt dem Analyse- statt dem Nachschlage-Bedarf entgegen. Sie unterscheidet sich von der ersten bezüglich des Inhaltes lediglich dadurch, daß die Doppel-Bedeutungen einiger Schlüsselwörter durch zwei verteilt angeordnete Zeilen anstelle einer einzigen Erwähnung berücksichtigt sind. Bezüglich der Reihenfolge der Schlüsselwörter wurde eine Einteilung in 6 Funktions-Gruppen gewählt; diese unterstützt u.a. auch einen Vergleich mit anderen Sprachen mit oder ohne derart umfangreiche Möglichkeiten für I/O und für Realzeitbetrieb, wie in PEARL konzipiert.

Die Funktions-Gruppen enthalten je die zugehörigen Objekte und Operationen; die Bezeichnungen der Gruppen lauten:

- I Allgemeine Strukturierung u. Verbindungen
- II Algorithmik
- III Standard-Operationen
- IV Sequentielle Programm-Abläufe
- V I/O - Kommunikation
- VI Parallele und Realzeit-Abläufe .

Innerhalb der Gruppen sind Untergruppen nach der Zugehörigkeit zu Basic PEARL oder der Ergänzung auf Full PEARL gebildet. Als Reihenfolge der Zeilen innerhalb der Untergruppen gilt alphabetische Folge der Schlüsselwörter, deren Kurzbeschreibungen beibehalten sind.

Unter den Auswertungs-Wegen zur T a b e l l e 4 unterscheidet sich eine Spalten-Analyse 2.1 im Prinzip nicht von 1.1. Das Gleiche gilt für die Analyse 2.2 gegenüber 1.2 (Zeilen-A.).

Als Auswertungs-Übersicht aus T a b e l l e 4 entstand T a b e l l e 5, und zwar zu Weg 2.3, d.h. verfeinert gegenüber 1.3. In jeder der Gruppen I bis VI bereits summiert sind die booleschen Verknüpfungs-Ergebnisse der "+"/"O" -

```
*****
TABELLE DER "+" / "0" -SUMMEN VON PEARL - SCHLUESSELWOERTERN
*****
```

	I M P L E M E N T I E R U N G E N											
	L	K	I	H	G	F	E	D	C	B	A	
SUMME "+"	108	133	139	118	125	133	116	91	85	83	130	
SUMME "0"	37	12	6	27	20	12	29	54	60	62	15	
GESAMT-SUMME	145	145	145	145	145	145	145	145	145	145	145	

Tabelle 2. Spaltensummen für Aussagen "+" bzw. "0" in Tabelle 1 zu den Implementierungen A bis L

\*\*\*\*\*  
TABELLE DER "+" / "0" -KOMBINATIONEN VON PEARL - SCHLUESSELWOERTERN  
SAEMTLICHER IMPLEMENTIERUNGEN (A BIS K)

KLASSE	BINAERMUSTER FUER SAEMTLICHE IMPLEMENTIERUNGEN		BASIC PEARL ? "J"	"N"		S U M M E N
ALPHA)	EINHEITLICH '++++++'		59	0		59 40 %
BETA )	MINDESTENS EIN '+' ODER EIN '0'		58	27		85 59 %
GAMMA)	EINHEITLICH '0000000'		0	1		1 1 %
S U M M E			117	28		145 100 %

Tabelle 3. Besetzung der Klassen ALPHA bis GAMMA von Binärkombinationen aus Aussagen "+" bzw. "0", zeilenweise in Tabelle 1 zu den Implementierungen A bis L enthalten

Aussage je Implementierungs-Spalte mit der "J"/"N" -Aussage in der Spalte "Basic PEARL ?" der selben Zeile. Diese Verknüpfungs-Ergebnisse sind in einer Vierfelder-Submatrix entweder in der einen Diagonalen als 'Äquivalenzen' zusammengehörend oder als 'Antivalenzen' in der anderen Diagonalen angeordnet. Beispielsweise zeigt sich in Gruppe V der Implementierung D die Ist-Besetzung mit 18 + 6 in der 'Äquivalenz'-Diagonalen gegenüber der Soll-Besetzung mit 34 + 9. Die restlichen 19 Fälle liegen in der 'Antivalenz'-Diagonalen, und zwar 16 im "schlechter als äquivalent" - Feld, entsprechend den "J" & "0" -Fällen sowie 3 im "besser als äquivalent" - Feld, entsprechend den "N" & "+" -Fällen. Diese Tabelle 5 bietet in der Übersicht nunmehr einige Informationen darüber an, welche der Implementierungen in welcher der Gruppen I bis VI eine zahlenmäßige Unterdeckung des Wortschatzes von Basic PEARL aufweist oder eine Volldeckung bzw. eine Überdeckung. Mit Ausnahme von B, C und D scheint es für die Imple-

mentierungen in den Gruppen I bis VI keine gravierenden Defekt-Fälle bezüglich Basic PEARL zu geben. Die Summen-Übersicht im unteren Teil der Tabelle gibt zusätzlich einige Hinweise, die zu Fragen der Portabilität pauschale Bedenken signalisieren. Für eine genauere Analyse ist es anhand von Tabelle 4 möglich, Abweichungen bei bestimmten Schlüsselwörtern, bzw. -Paaren zeilenweise zu vergleichen und diese in der Verwendung einzuschränken, z.B. wenn es sich um bibliotheksfähige Programm-Einheiten handelt. Als Beispiele hierfür erscheinen REENT und FOR-BACK.

Eine häufig anzutreffende Aussage: "die meisten Implementierungen gehen über Basic PEARL hinaus" läßt bezüglich der Tabelle 4 in den Zeilen mit einem "N" rechts von dem ~~F~~-Symbol die nähere Betrachtung besonders ratsam erscheinen; Beispiele: REF und CONT.

Dies könnte eine gegenüber dem Auswertungsweg 2.2 verfeinerte Untersuchung der Binärkombinationen je Zeile in der Klasse Beta wünschen lassen. Dabei wären Beta-Fälle zu unterscheiden, in

\*\*\*\*\*  
 GRUPPIERTES VERZEICHNIS DER SCHLUESSELWOERTER VON P E A R L (NACH FKT.-GRUPPEN)  
 \*\*\*\*\*

IMPLEMENTIERUNGEN		SCHLUESSELWORT	BEDEUTUNG , FUNKTION
:L:K:I:H:G:F:E:D:C:B:A		#??	
+++++		#JJDECLARE (DCL)	DEKLARAT.D.EIGENSCH.V.PEARL-GROESSEN
+++++		#J END	BLOCKENDE
+++++		#J GLOBAL	GLOB. GUELTIGK.VEREINB. ZWISCH.MODULN
+++++		#J MODEND	MODUL-ENDE
+++++		#J MODULE	MODUL-ANFANG
+++++		#J PROBLEM	ANFANG PROBLEM-TEIL
+++++		#JJSPECIFY (SPC)	SPEZIFIK.V.BEREITS DEKL.PEARL-GROESS.
:0:0:0:0:0:0:0:0:0:0:0		#N INLINE	DIREKT-EINBETTUNG V.CODE
:L:K:I:H:G:F:E:D:C:B:A		#??SCHLUESSELWORT	BEDEUTUNG , FUNKTION
+++++		#J BIT	DATENTYP: BITKETTE \ / SELEKTOR (.BIT)
+++++		#JJCHARACTER(CHAR)	DATENTYP: ZEICHENKETTE\ / SELEKT. (.CHAR)
+++++		#J CLOCK	DATENTYP: ZEITPUNKT (=UHRZEIT)
+++++		#JJURATION (DUR)	DATENTYP: DAUER, ZEITSPANNE
+++++		#J FIXED	DATENTYP: GANZZAHL\ / DATION-TYP (FILE)
+++++		#J FLOAT	DATENTYP: REELLE ZAHL
+++++		#J HRS	ZEITEINHEIT: STUNDE
:0+++++		#NJINITIAL (INIT)	ANFANGSWERT-ZUWEISUNG
+++++		#J INV	KONSTANT., INVARIABLE (-DEKLARAT.)
+++++		#J LENGTH	LAENGE (-DEKLARAT.)
+++++		#J MIN	ZEITEINHEIT: MINUTE
+++++		#J SEC	ZEITEINHEIT: SEKUNDE
+++++		#J STRUCT	STRUKTUR, VERBUND (-VEREINBARG.)
:0+++++		#N CONT	DEREFERENZIEREN EIN. REF-OBJEKTES
:0+++++		#N IS	VERGL.OPERAT. (F.REFER.) AUF 'TRUE'
:0+++++		#N ISNT	VERGL.OPERAT. (F.REFER.) AUF 'FALSE'
:0+++++		#N NIL	LEERE REFERENZ
:0+++++		#N OPERATOR	NICHT-STANDARD-OPERATOR (-VEREINBARG.)
:0+++++		#N PRECEDENCE	BENUTZ.DEF.VORRANG(-VEREINB.)F.OPERAT.
:0+++++		#N REF	REFERENZOBJEKT (-VEREINBARG.)
:0+++++		#N TYPE	NEUER-TYP -DEFINITION
:L:K:I:H:G:F:E:D:C:B:A		# STD.OPERATOR	BEDEUTUNG , FUNKTION
+++++		#J ABS	ABSOLUT-BETRAG
+++++		#J AND	LOG. KONJUNKTION (UND)
+++++		#J CAT	VERKETTUNG (CATENATION) V.CHAR OD. BIT
+++++		#J CSHIFT	KREISSCHIFT
+++++		#J ENTIER	GROESSTER GANZZAHLIGER WERT
+++++		#J EQ	VERGLEICH AUF "GLEICH"
+++++		#J EXOR	LOG. ANTIVALENZ (EXKLUSIV ODER)
+++++		#J FIT	LAENGEN-ANPASSUNG, FIXED OD. FLOAT
+++++		#J GE	VERGLEICH AUF "GROESSER ALS/GLEICH"
+++++		#J GT	VERGLEICH AUF "GROESSER ALS"
+++++		#J LE	VERGLEICH AUF "KLEINER ALS/GLEICH"
+++++		#J LT	VERGLEICH AUF "KLEINER ALS"
+++++		#J NE	VERGLEICH AUF "UNGLEICH"
+++++		#J NOT	LOG. NEGATION (BITWEISE INVERTIERG.)
+++++		#J OR	LOG. DISJUNKTION (INCLUSIV ODER)
:0+++++		#J ROUND	RUNDUNG
+++++		#J SHIFT	GESTRECKTER SCHIFT
+++++		#J SIGN	VORZEICHEN-BESTIMMUNG
+++++		#J TOBIT	TYPWANDLG.: FIXED IN BIT
+++++		#J TOCHAR	TYPWANDLG.: FIXED IN CHAR (ASCII)
+++++		#J TOFIXED	TYPWANDLG.: BIT ODER CHAR IN FIXED
+++++		#J TOFLOAT	TYPWANDLG.: FIXED IN FLOAT
+++++		#J UPB	OBERGRENZE V. ARRAY-INDIZES
:0+++++		#N LWB	UNTERGRENZE V. ARRAY-INDIZES
:0+++++		#N REM	REST NACH GANZZAHLIGER DIVISION

:L:K:I:H:G:F:E:D:C:B:A	???SCHLUESSELWORT	BEDEUTUNG , FUNKTION
+++++0+	#J ALT	ALTERNAT. BEI FALL-UNTERSCHIEDIG.(CASE)
+++++0++	#J BEGIN	BLOCKBILDUNGS-ANWEISUNG
+++++0++	#J BY	LAUFANW.SCHRITTW.\ / I/O-FORMAT-SPEZIF.
+++++0++	#J CALL	PROZEDUR-AUFRUF
+++++0++	#J CASE	FALL-UNTERSCHIEDIG(=SPRUNG-VERTEILER)
+++++0++	#J ELSE	"SONST"
+++++0++	#J END	BLOCKENDE
+++++0++	#J ENTRY	SPEZIFIK.E.PROZED.\ / TEIL V.INTFAC-DEKL.
+++++0++	#J FIN	ENDE "WENN"-BLOCK, CASE-ABSCHLUSS
+++++0++	#J FOR	LAUFLISTEN-ANFANG
+++++0++	#J FROM	GRENZE LAUFBEREICH\ / QUELLE F.EING.ANW.
+++++0++	#J GOTO	SPRUNG-ANWEISG.
0++++0+	#NJIDENTICAL(IDENT)	VEREINBARG. PROZEDUR-PARAMETER
+++++0++	#J IF	BEGINN "WENN"-BLOCK
+++++0++	#J OUT	UEBERTR.RICHT.;AUSG.\ / FEHL.AUSG.B.CASE
+++++0++	#JJPROCEDURE(PROC)	PROZEDUR (-VEREINB.)
++00++00+	#J REENT	MEHRF. BENUTZBARKEIT E.PROZEDUR
+++++0++	#J REPEAT	ANFANG LAUFBEREICH (WIEDERHOLGS.TEIL)
++00++00+	#J RESIDENT	AUSLAGERUNGSFAEHIG,JEDOCH SCHNELLZUGR.
+++++0++	#J RETURN	RUECKSPRUNG-ANWEISG. AUS PROZEDUR
+++++0++	#J RETURNS	TYP U.DIMENS. F.ERGBN.AUSG.E.PROZEDUR
+++++0++	#J THEN	"DANN"
+++++0++	#J TO	GRENZE LAUFBER. \ / ZIEL F.AUSG.ANWEIS.
+++++0++	#J WHILE	"SOLANGE ...GILT"
:L:K:I:H:G:F:E:D:C:B:A	???SCHLUESSELWORT	BEDEUTUNG , FUNKTION
:L:K:I:H:G:F:E:D:C:B:A	???SCHLUESSELWORT	BEDEUTUNG , FUNKTION
+++++0++	#J ALL	PERIOD.ZEITDAUER\ / CONTROL:ALLE FORMATE
+++++00+	#J ALPHIC	DAT.-,INTFAC.-TYP:DRUCKB. ZEICH.(ASCII)
+++++00+	#J BASIC	DAT.-,INTFAC.-TYP:BINAER-WERTE-KOMBIN.
+++++00+	#J BY	LAUFANW.SCHRITTW.\ / I/O-FORMAT-SPEZIF.
+++++00+	#J CLOSE	SCHLIESSEN EINER DATION
+++++0000+	#J CONTROL	STEUERGROESS.-VEREINBAR. E.DATION
+++++0000+	#J CREATED	BENUTZ.DEFIN. E.DATEI AUF DATION
0++++0000+	#J CYCLIC	FORTSCHALTG.-ATTRIBUT EIN.DATION
+++++000+	#J DATION	DATENSTATION (-VEREINBARUNG)
+++++0000+	#J DIM	DIMENSIONIERG.(-VEREINB.)F.DATION
0++++000+	#J DIRECT	RANDOM-ZUGRIFF AUF ELEM.E.DATION
+++++0000+	#J ENTRY	SPEZIFIK.E.PROZED.\ / TEIL V.INTFAC-DEKL.
+++++0000+	#J FIXED	DATENTYP: GANZZAHL\ / DATION-TYP (FILE)
++00++0000	#J FORBACK	SEQU.ZUGRIFF, VORW.OB.RUECKW.,A.DATION
+++++0000+	#J FORMAT	DEFINIT. EIN. FORMAT-STEUERLISTE
+++++0000+	#J FORWARD	SEQUENT.ZUGRIFF, VORWAERTS, AUF DATION
+++++0000+	#J FROM	GRENZE LAUFBEREICH\ / QUELLE F.EING.ANW.
+++++0000+	#J GET	EINGABE-ANWEISG. F.ALPHIC-DARSTELLG.
+++++0000+	#J IN	UEBERTRAGUNGSRICHTG.: EINGABE
+++++0000+	#J INOUT	BEIDE UEBERTRAGUNGSRICHTUNGEN
+++++0000+	#J MAX	MAX.LAENG.ANGABE Z.UEBERTRAG.EINH.TFU
00++00++0000+	#J NOCYCL	GEGENTEIL,FORTSCHALT.ATTRIB.ZU CYCLIC
00++00++0000+	#J NOSTREAM	GEGENTEIL,FORTSCHALT.ATTRIB.ZU STREAM
+++++0000+	#J OPEN	OEFFNEN E.DATION \ / TEIL E.INTFAC-DEKL.
+++++0000+	#J OUT	UEBERTR.RICHT.;AUSG.\ / FEHL.AUSG.B.CASE
+++++0000+	#J PUT	AUSGABE-ANWEISG. F.ALPHIC-DARSTELLG.
+++++0000+	#J READ	EINGABE-ANWEISG. F.INTERN-DARSTELLG.
+++++0000+	#J SEND	AUSGABE-ANWSG. F.BASIC-DARSTELLG.
00++00++0000+	#J STREAM	FORTSCHALT.-ATTRIBUT E.DAT.STAT.
+++++0000+	#J SYSTEM	ANFANG SYSTEM-TEIL
+++++0000+	#J TAKE	EINGABE-ANWEISG. F.BASIC-DARSTELLG.
+++++00000+	#J TFU	UEBERTRAGUNGSEINHEIT F.DAT.KANAL
+++++0000+	#J TO	GRENZE LAUFBER. \ / ZIEL F.AUSG.ANWEIS.
++00++0000+	#J WRITE	AUSGABE-ANWEISG. F.INTERN-DARSTELLG.
0000++00000++	#N CREATE	KREIEREN EINER DATION
0000++00000++	#N DELETE	ZERSTOER.V.NICHT MEHR BENDET.DATION
0000++00000000	#N EXIT	TEIL V.INTFAC-DEKLARAT.
0000++000000000	#N INTFAC	SOFTW.INTFAC (N.-STAND.)ZWISCH.DATIONS
0000++000000000	#N MATCH	INTFAC-PARAM. F.STEUERKANAL
0000++0000000000	#N NOMATCH	GEGENTEIL. INTFAC-PARAM. ZU MATCH
0000++000000000+	#N ONEOF	TYP UEBERTRAG.DATEN (=EIN.AUS TYPLISTE
0000++00000000++	#N UPON	UNTERE DATION (=GERAET) BEI CREATE
0000++0000000000	#N USING	INTFAC-VERWENDG. BEI CREATE E.DATION

Tabelle 4 - Teil 2

:L:K:I:H:G:F:E:D:C:B:A	??SCHLUESSELWORT	BEDEUTUNG , FUNKTION
+++++	#J ACTIVATE	STARTEN EINER TASK
+++++	#J AFTER	START-BEDINGUNG: ZEITSPANNE
+++++	#J ALL	PERIOD.ZEITDAUER/CONTROL:ALLE FORMATE
+++++	#J AT	START-BEDINGUNG: ZEITPUNKT
+++++	#J CONTINUE	FORTSETZEN E.TASK-AKTIVITAET
+++++	#J DISABLE	ABSCHALTUNG EIN.INTERRUPTS
+++++	#J DURING	DAUER BIS ENDE(DANACH KEINE WIEDERHOL.)
+++++	#J ENABLE	FREIGABE E.INTERRUPTS
0++++	#J INDUCE	EXPLIZ.AUSLOES.E.AUSNAHMEBEDING.(SIGN.)
+++++	#JJINTERRUPT(IRPT)	UNTERBRECHUNG (-VEREINB.)
0++++	#J ON	REAKT.DEFINIT.AUF SIGNAL(=AUSN.BEDING.)
+++++	#J PRESET	ANFANGSWERT-ZUWEISG.F.SEMA-VAR.,-ARRAY
+++++	#J PREVENT	AUSPLANEN E.TASK-AKTIVITAET
+++++	#JJPRIORITY(PRIO)	TASK-PRIORITAET (IN PEARL-MODUL !)
+++++	#J RELEASE	EXPLIZ.FREIGABE E. SEMA-VARIABLEN
+++++	#J REQUEST	EXPLIZ.SPERRUNG E.SEMA-VARIABLEN
+++++	#J RESUME	VERZOEG.FREIGABE E.TASK-AKTIVITAET
+++++	#J SEMA	SEMAPHOR-VARIABLE
0++++	#J SIGNAL	REAKTION(-VEREINB.) AUF LAUFZEITFEHLER
+++++	#J SUSPEND	ANHALTEN EIN.TASK-AKTIVITAET
+++++	#J TASK	TASK (-VEREINBARG.)
+++++	#J TERMINATE	(VORZEITIGES) BEENDEN E.TASK-AKTIVIT.
+++++	#J UNTIL	ZEITPUNKT F.ENDE V.TASK-WIEDERHOLGN.
+++++	#J WHEN	TASK-STARTBEDING.: ABHAENG.V.INTERRUPT
0++++	#N BOLT	BOLT-VARIABLE
0++++	#N ENTER	BEGINN SIMULT.ANFORDERG.(BOLT-VARIABLE)
+++++	#N EVERY	START-BEDING.:PERIOD.ZEITPKT.(VOLLE E.)
0++++	#N FREE	ENDE E.EXKL.BENUTZG.(FREIGABE) BOLT-V.
0++++	#N LEAVE	ENDE SIMULT.ANFORDERG.(BOLT-VARIABLE)
0++++	#N RESERVE	EXCLUS.BENUTZG. (=SPERRE) E.BOLT-VAR.
00++++	#N SYS	SYSTEM-REAKTION AUF ON-(SIGN.)ANWSG.
+++++	#N TRIGGER	AUSLOESEN E.INTERRUPTS
:L:K:I:H:G:F:E:D:C:B:A	??SCHLUESSELWORT	BEDEUTUNG , FUNKTION

Tabelle 4. Gruppiertes Verzeichnis der Schlüsselwörter von PEARL, entsprechend Tabelle 1 umgeordnet nach Funktions-Gruppen I bis VI (von oben nach unten)

TABELLE DER '+ / 0' TEILSUMMEN VON PEARL - SCHLUESSELWORTERN IN DEN SPRACHTEILEN UND FKTS-GRUPPEN	
GRUPPE	BASIC : L : K : I : H : G : F : E : D : C : B : A : SOLL
I	SUMME 'J' : 7 0 7 0 7 0 7 0 7 0 7 0 6 0 1 7 0 SUMME 'N' : 1 0 1 0 1 0 1 0 1 0 1 0 1 1 0 1 0 1
II	SUMME 'J' : 13 0 13 0 13 0 13 0 13 0 11 2 10 3 11 2 13 0 SUMME 'N' : 8 0 6 2 0 8 1 7 8 0 8 0 8 0 6 2 0 8
III	SUMME 'J' : 23 0 23 0 23 0 23 0 23 0 15 8 20 3 12 11 23 0 SUMME 'N' : 2 0 2 0 2 0 1 1 2 0 2 0 2 1 1 0 2 0 2
IV	SUMME 'J' : 24 0 22 2 23 1 24 0 24 0 24 0 24 3 17 7 24 0 SUMME 'N' : 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
V	SUMME 'J' : 34 0 31 3 34 0 33 1 34 0 33 1 18 16 16 21 13 33 1 SUMME 'N' : 0 9 9 0 0 9 0 9 0 9 0 9 3 6 0 9 3 6 4 5 0 9
VI	SUMME 'J' : 24 0 24 0 24 0 24 0 24 0 20 4 16 8 20 4 24 0 SUMME 'N' : 6 2 8 0 8 7 1 6 2 0 8 1 7 0 8 1 7 2 6 0 8
	SUMME 'J' : 125 0 120 5 124 1 124 1 125 0 124 1 95 30 92 33 87 38 124 1 125 0 GESAMT SUMME 'N' : 16 12 27 1 2 26 9 19 16 12 0 28 4 24 1 27 4 24 14 0 28 0

T a b e l l e 5

denen zwischen einer einzigen und sämtlichen (minus 1) "+" -Aussagen als 'Quersumme' möglich sind. Diese Fälle können z.B. durch eine Angabe wie BETA4+ für vier Implementierungen mit "+", sowie dem Rest mit "0", gekennzeichnet werden; Beispiele: CREATE, BOLT. Hierauf sei an dieser Stelle verzichtet.

Spätestens hier ist jedoch zu berücksichtigen, daß unter den zugrundegelegten Implementierungen insgesamt keine Gleichheit der Anwendungs-Breite und -Ziele sowie der ausführenden Rechner besteht. Die Auswahl wurde nach /3/ getroffen. Die entschlüsselten Bezeichnungen zu den Implementierungen lauten:



L	K	I	H	G	F	E	D	C	B	A	Compilerhersteller:
.	.	.	.	.	.	.	.	.	.	.	--ATM Computer / AEG
.	.	.	.	.	.	.	.	.	.	.	----BBC & Cie
.	.	.	.	.	.	.	.	.	.	.	-----Dornier System
.	.	.	.	.	.	.	.	.	.	.	-----Reg.RZ UNI Erlangen
.	.	.	.	.	.	.	.	.	.	.	-----GPP, Ges.f.Pr.R.Progr.
.	.	.	.	.	.	.	.	.	.	.	-----Entw.Büro Werum
.	.	.	.	.	.	.	.	.	.	.	-----Krupp Atlas-Elektronik
.	.	.	.	.	.	.	.	.	.	.	-----Siemens
.	.	.	.	.	.	.	.	.	.	.	-----Digital Equipment
.	.	.	.	.	.	.	.	.	.	.	-----Fraunhofer-Inst.IITB
.	.	.	.	.	.	.	.	.	.	.	-----Inst.Reg.T. UNI Hannov.

Portable PEARL-Systeme für mehrere sowie unterschiedliche Übersetzungs- und Ziel-Rechner liegen z.B. unter E und F vor. K ist mit F verwandt, verfügt jedoch zusätzlich über Spracherweiterungen für verteilte Systeme. Zu den übrigen genannten PEARL-Übersetzungssystemen wurden Charakterisierungen in mehreren früheren Beiträgen dieser Zeitschrift bereits geboten, wie im übrigen auch für E, F und K.

Für den allgemeinen Anwender erschöpfen sich die Probleme der 'Querbeet'-Portabilität nicht in den hier behandelten Schlüsselwort-Problemen. Die syntaktischen Bestandteile des Sprachumfanges können den Programmierkomfort und die Effizienz erhöhen, sind jedoch nicht bei allen Implementierungen durchgehend verfügbar. Stichwortartige Beschreibungen von Einschränkungen bei konkreten Implementierungen gegenüber den beiden Teilsprachen werden beispielsweise in /4/ und /5/ offengelegt.

Es bestehen jedoch auch sehr elementare sowie zeitraubende Hindernisse der Übertragbarkeit von PEARL-Moduln bereits darin, daß z.B. bei einigen Compilern ein Modul-Bezeichner angegeben werden kann oder muß oder nicht darf /6/.

Über die gegebenen Interpretations-Hinweise hinaus waren in dem beabsichtigten Rahmen dieses Beitrages weder ausdrückliche Bewertungen der PEARL-Implementierungen, noch Sprachvergleiche, z.B. ähnlich /7/ und /8/, vorgesehen.

Eine Einteilung des PEARL-Sprachumfanges in mehrere Stufen wurde in /9/ verwendet; hierzu lauten die Stichworte:

- Initialisieren von Arrays u. STRUCTURES,
- erweiterte String-Selektion,
- erweiterte Ausführungsbeding.(Schedules),
- Prozedur-Deklarationen auf allen Ebenen,
- bedingte Ausdrücke,
- REFERenzen auf Objekte,
- Operatoren von Full PEARL,
- Deklarationen von neuen Datentypen,

- Deklarationen von Operatoren,
- BOLTs,
- Interfaces ohne s/i-channel,
- " mit " " "

Die tabellarische Übersicht in /10/ zu den für Anwender interessanten Eigenschaften von PEARL-Systemen, die gegenwärtig am Markt existieren, ergänzt die hier getroffene Auswahl der Implementierungen A sowie C bis K .

Schrifttum

/1/ DIN 66253 / Teil 1 Basic PEARL, Vornorm 1981

/2/ DIN 66253 / Teil 2 Full PEARL, Normentwurf

/3/ Übersicht: PEARL - Compiler  
PEARL Mitglieder-Informationen 1982 / Nr.3

/4/ Werum / Windauer: PEARL  
Vieweg Verlag, Braunschweig, 1978

/5/ PEARL für Computersysteme ATM 80, Sprachbeschreibung, ATM Computer GmbH

/6/ Röhrich: Portabilitäts-Erfahrungen mit PEARL, demonstriert am Beispiel des Analogdaten-Erfassungs-Moduls an verschiedenen Prozeßrechnern.  
Referat 21.9.82 Universität Stuttgart, Institut f.Verfahrenstechnik u.Dampfkess.Wes. bei Vorstellung PEARL-Anwender-Progr.Bibl.

/7/ König / Schwarz: Sprachvergleich der FORTRAN-Compiler.  
Zeitschr.f.Datenverarbeitung/ONLINE, 1973

/8/ Simon: Vergleiche der Sprachmengen der COBOL-Compiler.  
Zeitschr.f.Datenverarbeitung, 1972

/9/ Rzehak / Heilbrunner: Vortrag über Vergleich von PEARL-Kompilier-Systemen.  
Ergebn.Bericht zur Sitzg. d.PEARL-Anwender-Ausschußes am 26.5.82

/10/ Weber: Übersicht über PEARL-Umgebungen, Eigenschaften von PEARL-Systemen.  
PEARL Mitglieder-Informationen 1982 / Nr.4

Anschrift des Autors:

Prof. Horst Meintzen  
Gallus-Zembrod-Str. 23  
7753 Allensbach 1

