

EDV-Terminologie (Folge 5)

ALGOL

(ALGOritmic Language)

ALGOL ist eine problemorientierte höhere Programmiersprache, die hauptsächlich für den Einsatz im mathematisch-wissenschaftlichen Bereich entwickelt wurde. Sie weist starke Ähnlichkeit mit der üblichen mathematischen Schreibweise auf. (Daher auch der ursprüngliche Name IAL für „International Algebraic Language“.) ALGOL wurde in Europa entwickelt, als man in den USA an → FORTRAN arbeitete. Das Konzept beim Entwurf dieser Sprache war, im → Quellcode die Möglichkeit zu bieten, Algorithmen (→ Algorithmus) durch die Abfolge der Befehle und Deklarationen klar hervortreten zu lassen (daher der endgültige Name ALGOL).

Verschiedene Eigenschaften von ALGOL sind in die Programmiersprache Pascal übernommen worden, so vor allem die → Blockstruktur. Diese Struktur erlaubt es, ein Programm in mehrere Blöcke mit genau definierten Wechselwirkungen einzuteilen. Dadurch ist es möglich, daß mehrere Programmierer arbeitsteilig an einem Programm arbeiten. Auch wird die Wartung (→ Programm, Wartung) so strukturierter Programme erleichtert.

APL

(A Programming Language)

Es handelt sich um eine Programmiersprache, die insbesondere für die interaktive Bearbeitung mathematischer Probleme bestimmt ist. Das Bemerkenswerte an dieser Sprache, die 1962 von Iverson definiert wurde, ist die Verwendung eines eigenen APL-Zeichensatzes, der die in dieser Sprache erstellten Programme als sehr kompakt erscheinen läßt. Eine Konsequenz der Wahl dieses speziellen Zeichensatzes ist es, daß APL nur in Verbindung mit Terminals verwandt werden kann, die in der Lage sind, den APL-Zeichensatz darzustellen. APL ist mittlerweile auch auf Mikrocomputern verfügbar. Besonders im Bereich der Statistik gibt es sehr leistungsfähige APL-Software.

BASIC

(Beginner's All Purpose Symbolic Instruction Code)

BASIC ist heute im Mikrocomputerbereich die wohl am weitesten verbreitete Programmiersprache. Das hat seinen Grund darin, daß die Anbieter von Mikrocomputern ihre Systeme meistens mit einem im Lieferumfang enthaltenen BASIC-Interpreter (→ Interpreter) ausstatten. BASIC ist 1965 am Dartmouth-College (USA) entwickelt worden. Die Sprache wurde ursprünglich nur für Lehrzwecke eingesetzt. Die Grundversion kennt keine Strukturierungselemente (→ Programmierung, strukturierte).

BASIC ist nur im Grundwortschatz weitgehend einheitlich definiert. Zusatzdefinitionen der einzelnen Hersteller haben zu einer Sprachentwicklung geführt, die heute durch die gleichzeitige Existenz von verschiedenen „BASIC-Dialekten“ gekennzeichnet ist. Dies gilt insbesondere für Erweiterungen der Sprache um Strukturierungselemente und Befehle zur Prozedurdeklaration (→ Prozedur). Zweck dieser Erweiterungen ist es, dem allgemein als Mangel empfundenen Fehlen von Strukturierungselementen in den ursprünglichen BASIC-Versionen abzuwehren. Der Grundwortschatz, der in praktisch allen BASIC-Implementationen (→ Implementation) weitgehend gleich ist, umfaßt die Befehle DATA, END, GOTO, INPUT, LET, PRINT. Man sieht daraus, daß der gemeinsame Kern aller „BASIC-Dialekte“ recht klein ist. Normungsbestrebungen haben noch nicht zu einer Abhilfe geführt. Eine Vereinheitlichung zeichnet sich aber faktisch durch die weite Verbreitung der von der Firma Microsoft entwickelten BASIC-Variante auf IBM-PC's und dazu kompatiblen Rechnern ab.

C

Die Programmiersprache „C“ ist in letzter Zeit zunehmend populär geworden. Sie ist im Zusammenhang mit der Entwicklung des Betriebssystems → UNIX entstanden und diente lange Zeit hauptsächlich der Programmierung von Betriebssystemen (→ Betriebssystem).

Die Sprache „C“ zeichnet sich durch sehr einfache und maschinennahe Befehle aus und enthält z.B. keinerlei Befehle für Ein- oder Ausgabeoperationen. Diese Funktionen müssen in einer speziellen Bibliothek als Prozeduren zur Verfügung gestellt werden. Auf diesem Konzept beruht die leichte → Portabilität von in „C“ geschriebenen Programmen. Lediglich einzelne maschinenabhängige Programmteile müssen beim Wechsel von einem System zum anderen neu programmiert werden. Die Programmiersprache „C“ entfaltet ihre volle Bedeutung erst in Verbindung mit der sogenannten „UNIX-Library“, die dem „C“-Programmierer bestimmte Funktionen aus dem UNIX-Betriebssystem zur Verfügung stellt. Aber auch auf Systemen, die nicht das Betriebssystem UNIX verwenden, steht mittlerweile diese Bibliothek mit leistungsfähigen Standardfunktionen zur Verfügung. Teilweise gehört sie bereits zum Lieferumfang des C-Compilers (→ Compiler). C ist noch nicht standardisiert, es gibt aber einen vorgeschlagenen ANSI-Standard.

COBOL

(Common Business Oriented Language)

Es handelt sich um eine der ältesten Programmiersprachen. Sie wurde bereits 1959 für die überwiegend

aus Ein- und Ausgabevorgängen bestehenden Probleme der kommerziellen Datenverarbeitung entwickelt. COBOL gehört zu den wenigen genormten Sprachen, für die eine einheitliche Definition des Sprachstandards existiert. Die „Conference on Data Systems Languages“ (CODASYL) hat 1960 den Umfang der Version COBOL-60 festgelegt. Das „American National Standards Institute“ (→ ANSI) definierte 1974 das sogenannte ANSI-COBOL-74. An diese Definition ist auch der DIN-Entwurf 66028 „Programmiersprache COBOL“ angelehnt. Neben den genannten Normen existiert zusätzlich noch eine ISO-Norm.

Hauptproblem bei COBOL ist die Festlegung des Sprachumfangs in zwei Ebenen. So kann ein → Compiler dem Standard auf der höchsten Ebene entsprechen und die mit diesem Compiler erstellten Programme können mit einem Compiler, der nur die unterste Normstufe realisiert, nicht kompiliert werden.

Der Hauptvorteil von COBOL ist, daß die Sprach-elemente in didaktisch günstiger Form der englischen Sprache entlehnt sind. Deshalb liest sich ein Programm im → Quellcode teilweise wie ein normaler englischer Text (z. B. „MOVE 3 TO ZAHL“ für die Zuweisung des Wertes 3 zu der Variablen „Zahl“ oder „MULTIPLY QUANTITY BY PRICE GIVING TOTAL PRICE“ für die Multiplikation von Preis mit Anzahl zur Errechnung des Gesamtpreises). COBOL ist daher nach allgemeiner Meinung leicht zu lernen. Auch sind COBOL-Programme für unterschiedliche Programmierer leicht lesbar, was die Arbeitsteilung und Kooperation erleichtert. Schließlich ist die Dokumentation leichter zu erstellen.

Auf Microcomputern ist COBOL heute wegen der in dieser Sprache geschriebenen umfangreichen Anwendungsprogramme ebenfalls weit verbreitet. Hier wurde der COBOL-Standard von verschiedenen Herstellern um Module zur Bildschirmmaskenbearbeitung (→ Bildschirmmaske) erweitert, da die ursprünglichen Compiler nur für → Batch-Datenverarbeitung konzipiert waren. Diese Ergänzung bewegt sich aber außerhalb des Standards, so daß dadurch die → Portabilität stark eingeschränkt wird. Ohne derartige Erweiterungen sind COBOL-Programme mit geringem Aufwand von Maschine zu Maschine übertragbar, da der geräteabhängige Teil eines Programms in der „Environment-Division“ zusammengefaßt ist.

FORTRAN

(FORMula TRANslator)

Diese problemorientierte höhere Programmiersprache entstand Mitte der fünfziger Jahre in den Entwicklungslabors der Firma IBM. Das Ziel des von J.W. Backus geleiteten Teams war, eine Programmiersprache für mathematische und naturwissenschaftliche Anwendungen zu schaffen. Dabei sollte gleichzeitig durch die Kombination von Formelschreibweise und an das Englische angelehnten Befehlen eine leichte Lesbarkeit erreicht werden.

Der erste → Compiler (FORTRAN II) wurde bald von den auch heute noch weit verbreiteten FORTRAN-IV Compilern abgelöst. Ähnlich wie → COBOL ist auch FORTRAN von der → ANSI genormt, so daß ein einheitlicher Sprachstandard existiert. Dieser Standard wurde 1977 noch einmal erweitert. Das Ergebnis dieser Erweiterung ist FORTRAN-77. FORTRAN ist am besten zur Bearbeitung numerischer Probleme geeignet und kennt hier zahlreiche Datentypen (→ Integer, → Real, → Complex, → Logical, → Double Precision). Zur Verarbeitung von Texten und nicht numerischen Daten ist FORTRAN-IV ungeeignet. Erst FORTRAN-77 führt einen hierfür brauchbaren Datentyp (→ String) ein.

FORTRAN ist überwiegend im technisch-wissenschaftlichen Bereich im Einsatz und ergibt sehr gut portierbare (→ Portierung) Programme.

LISP

(LIST Processing Language)

LISP existiert seit fast 30 Jahren. Die Sprache wurde am Massachusetts Institute of Technology (MIT) entwickelt. LISP ist eine Programmiersprache, die hauptsächlich in Zusammenhang mit Forschungsprojekten im Bereich der „künstlichen Intelligenz“ (→ KI; Intelligenz, künstliche) eingesetzt wird. Das auch heute noch vielbesprochene Programm → ELIZA, das den Dialog eines Klienten mit einem Psychotherapeuten simulieren sollte, ist ursprünglich in LISP realisiert worden. LISP konnte lange Zeit wegen des hohen Hauptspeicherbedarfs nur auf Großrechnern wirkungsvoll eingesetzt werden. Seit einiger Zeit gibt es jedoch auch für Mini- und Microcomputer leistungsfähige LISP-Interpreter.

Wie bereits der Name sagt, ist die Grundstruktur dieser Programmiersprache die → „Liste“. Darunter versteht man einen aus Elementen und Operationsregeln bestehende komplexen Klammerausdruck, der schrittweise von innen nach außen ausgewertet wird. In dieser Struktur lassen sich vor allen Dingen rekursive (→ Rekursion) Operationen besonders gut abbilden. Da innerhalb von Problemlösungsstrategien die Rekursion eine besondere Rolle spielt, ist LISP für die Bearbeitung derartiger Fragestellungen sehr geeignet.

Das ist einer der Gründe für die weite Verbreitung von LISP im KI-Bereich. Ein weiterer ist darin zu sehen, daß LISP zur Gruppe der „deklarativen“ Programmiersprachen (→ Programmiersprachen, deklarative) gehört. Diese Sprachen sind im Unterschied zu den „befehlsorientierten“ Programmiersprachen (→ Programmiersprachen, befehlsorientierte) dadurch gekennzeichnet, daß im Anschluß an eine Beschreibung des Problems (im Rahmen der vorgegebenen Beschreibungsmöglichkeiten für die Problemstruktur) bestimmte Komponenten der Programmiersprache ohne Detailprogrammierung der einzelnen Schritte die denkbaren Problemlösungen erarbeiten.