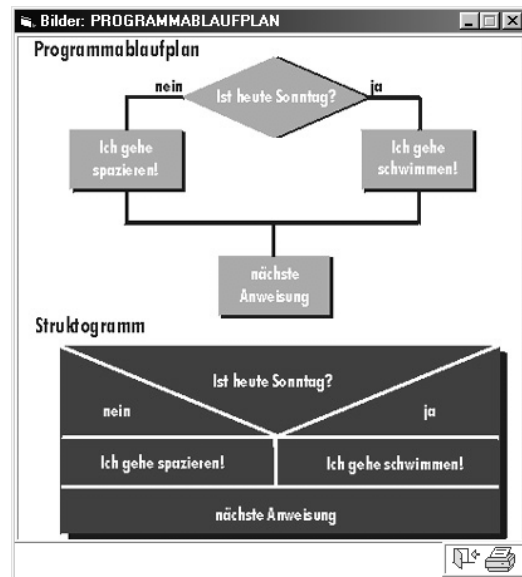


## Programmierung

Programmieren ist das Erstellen eines Programms oder den Teil eines Programms durch einen Programmierer. Unter Verwendung einer oder mehrerer Programmiersprachen werden Lösungen für eine Aufgabe gefunden. Zur Programmierung gehören die exakte Formulierung der Aufgabe und des Lösungsmodells und schließlich das eigentliche Codieren in den gewählten Programmiersprachen.

Zur Entwicklung eines Programms bedarf es fünf Arbeitsschritte:

1. Beschreibung eines Problems
2. Entwicklung einer Lösungsvorschrift
3. Übersetzung in eine oder mehrere Programmiersprachen
4. Fehlersuche und Test des Programms
5. Dokumentierung (Herstellung des Handbuches)



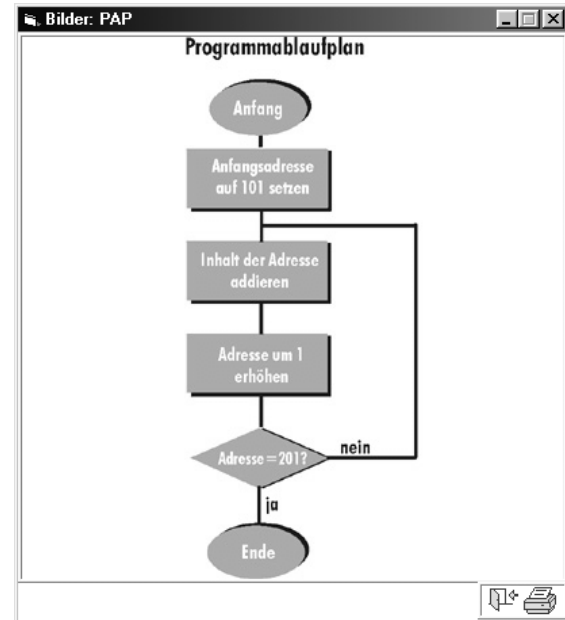
Vor der Erstellung der Befehlsabfolge für ein Programm macht sich der Programmierer einen Plan für sein Vorgehen.

## Programmiertechnik

Zur Programmiertechnik gehören die exakte Aufgabenformulierung und die Formulierung des Lösungsmodells, etwa in Form eines Struktogramms oder eines Programmablaufplans, die Festlegung auf eine oder mehrere Programmiersprachen, algorithmische Verfahren und Datenstrukturen und am Schluss das eigentliche Codieren. Dazu kommen die Beschreibung des Programms und die Programmtests wie Fehlerverhalten, Speicherbedarf und Laufzeit. Alle diese Schritte werden heute überwiegend als ingenieurmäßige Arbeit gesehen und müssen sorgfältig dokumentiert werden, vor allem das fertige Programm.

Programme werden nach der sprachlichen Ebene unterschieden, in der sie geschrieben wurden. Ein Quellprogramm ist meist in einer maschinenfernen, höheren Programmiersprache verfasst und muss durch einen Übersetzer in ein Maschinenprogramm übersetzt werden. Ein in höherer Programmiersprache vorliegendes Quell-Programm ist im allgemeinen auf unterschiedlichen Rechenanlagen zu verwenden, wobei Programme in Maschinensprache auf eine bestimmte Rechnerart angewiesen sind.

Ein Programmablaufplan enthält alle wichtigen Schritte, die ein Programm abarbeiten soll.



### Problembeschreibung

Wenn ein Programmierer ein Programm für einen Computer schreiben will, muss er sich erst einmal mit dem Problem, der Aufgabe, vertraut machen. Er muss wissen, welche Daten er wo und wie eingeben muss, wie sie weiter verarbeitet und wie sie ausgegeben werden sollen. Dazu benötigt er Angaben über die Anzahl und die Art der zu verwaltenden Daten. Erst jetzt kann er zum nächsten Arbeitsschritt, der Problembeschreibung, gehen. Für größere Programmsysteme wird hierfür sehr viel Zeit eingeräumt, die, und das lässt sich kaum vermeiden, durch die Arbeit eines System-Analytikers verkürzt werden kann.

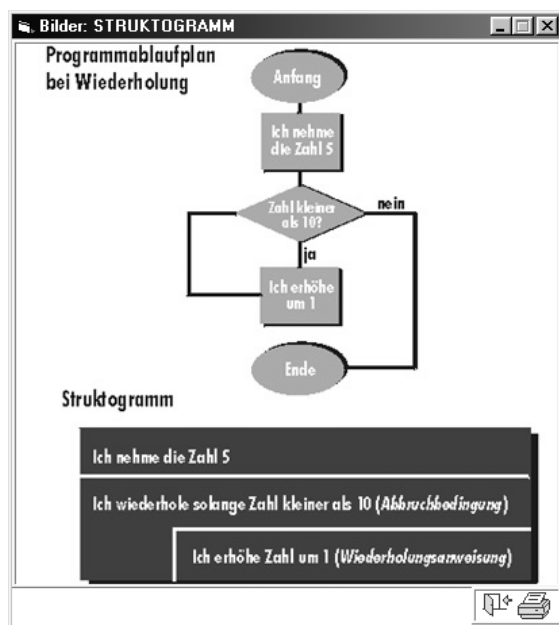
Die System-Analyse untersucht den Prozess, wie ein existierendes System arbeitet, welches die Grundlage zur Verbesserung des Programms ist. Der System-Analytiker muss bei seiner Analyse auch die Menschen, die mit dem System in Berührung kommen, berücksichtigen, denn das beste System nutzt nichts, wenn es vom Benutzer nicht akzeptiert wird.

### Lösungsvorschrift

Die Lösungsvorschrift soll dokumentieren, in welchen einzelnen Schritten ein Problem innerhalb der Programmieretechnik gelöst werden kann. Um eine überschaubare Dokumentation zu bekommen, benutzt man Diagramme, in dem Art und Aufeinanderfolge der zur Lösung einer Problemstellung mit Hilfe eines Computers erforderlichen Einzeloperationen dargestellt sind.

Grafische Hilfsmittel wie die am meisten angewendeten Programmablaufpläne und Struktogramme kennzeichnen mit speziellen (nach DIN genormten und unabhängig von der jeweiligen Programmiersprache) Symbolen die einzelnen Arbeitsschritte wie das Festsetzen einer oder mehrerer Programmiersprachen, die Festlegung auf algorithmische Verfahren sowie Datenstrukturen.

Die Verwendung des Struktogramms beim Programmentwurf zwingt zur disziplinierten, strukturierten Programmierung und ist deshalb dem Programmablaufplan vorzuziehen.



## Umsetzung und Test

Nach der Erstellung einer Lösungsvorschrift innerhalb der Programmierertechnik beginnt die Umsetzung. Die Arbeitsschritte werden in eine Programmiersprache übersetzt. Mehrere Programmiersprachen stehen dem Programmierer zur Verfügung, zum Beispiel: ADAL, ALGOL, BASIC, C und C++, COBOL, COMAL, ELAN, FORTRAN, Turbo PASCAL, SIMULA, SNOBOL, Visual Basic, Delphi.

Es hängt von den Kenntnissen des Programmierers und der Art des Problems ab, welche Programmiersprache zum Tragen kommt. Jede Programmiersprache ist nicht unbedingt für jedes Problem geeignet. In einem Editor, ähnlich einem herkömmlichen Textverarbeitungsprogramm, wird das Programm von einem Programmierer erstellt.

Nach Fertigstellung des Programms muss es nun auf Fehler untersucht werden, denn Fehler sind fast unvermeidbar. Der Programmtext wird durch ein Übersetzungsprogramm in die für den Computer verständliche Sprache übersetzt und meldet Tippfehler und auch solche, die gegen die Regeln der Programmiersprache verstoßen.

Um auch logische Fehler erkennen zu können, gibt es sogenannte Debugger, spezielle Programme, die das zu testende Programm Schritt für Schritt ausführen und das Ergebnis dem Programmierer mitteilen.

## **Programmiersprache**

Eine Programmiersprache ist eine zur Erstellung von Programmen geschaffene, streng formalisierte Sprache. Ihre Ausdrucksmittel sind entsprechend den zu bearbeitenden Problemkreisen gestaltet, jedoch frei von konkreten Zusammenhängen, man sagt "kontextfrei".

Heute gibt es etwa 200 Programmiersprachen, wobei noch die Sprachen dazu zu zählen sind, die im Laufe der Computerzeit für ganz bestimmte Zwecke entwickelt wurden. Kein Programmierer kann all diese Sprachen beherrschen. Um Programme auch auf andere Rechner übertragen zu können, haben sich einige Standards entwickelt.

Unabhängig von den zugrunde gelegten Zielen und bestimmten Einsatzgebieten haben Programmiersprachen unterschiedliche Eigenschaften, darum werden sie in fünf verschiedene Generationen unterteilt.

### Sprachgenerationen:

1. Generation: Maschinensprachen
2. Generation: Assemblersprachen
3. Generation: höhere Computersprachen
4. Generation: Makro-Sprachen
5. Generation: KI-Sprachen

Die Generationen sind als Hinweis darauf, in welcher Abfolge Sprachgruppen auf Rechenanlagen anwendbar sind, zu verstehen. Für jeden neuen Computer muss eine neue Maschinensprache, eine Sprache der ersten Generation entwickelt werden. Assemblersprachen der 2. Generation sind Hilfsmittel, um die Maschinenbefehle einprägsamer und besser verständlich darzustellen. Mangelhafte Eignung maschinenorientierter Sprachen zur Erstellung komplexer Anwendungsprogramme gaben den Anstoß zur Entwicklung "höherer" Programmiersprachen. Bei den Sprachen der 4. Generation handelt es sich um fertige Anwendungsprogramme, die mit Hilfe der integrierten Sprache Lösungen für ganz spezielle Probleme beschreiben können. Sprachen der 5. Generation: Darunter versteht man Computerprogramme, die geistige Fähigkeiten des Menschen simulieren können, sie sind meist mit dem Begriff "Künstliche Intelligenz" verbunden. Am verbreitetsten sind die Programmiersprachen: ADAL, ALGOL, BASIC, C und C++, COBOL, COMAL, ELAN, FORTRAN, Turbo PASCAL, SIMULA, SNOBOL, Delphi.

## **Visual Basic**

BASIC war in den Anfangszeiten des PCs eine der verbreitetsten Programmiersprachen für Heimanwender. Visual Basic ist eine auf Basic beruhende, objektorientierte Programmiersprache der Firma Microsoft, die speziell für die Programmierung unter

Windows entwickelt wurde. Heute gehört Visual Basic bei Profis und Hobby-Programmierern mit zu den beliebtesten Programmiersprachen, weil man in relativ kurzer Zeit ausgereifte Programme entwickeln kann, ohne über herausragendes Wissen bezüglich der verwendeten Programmiersprache verfügen zu müssen. Inzwischen ist Visual Basic selbst bei Anwendungen, bei denen schnell gerechnet werden muss, uneingeschränkt einsatzfähig, weil die neue Entwicklungsumgebung Visual Basic 5 auch die Komplimierung eines Programms ermöglicht.

Mit Visual Basic for Applications (VBA) darf an dieser Stelle eine weitere Variante nicht unerwähnt bleiben. Diese Programmiersprache wurde von Microsoft als Makrosprache für seine Anwendungen eingesetzt. Durch die hohe Leistungsfähigkeit und die Programm-übergreifenden Eigenschaften wird so eine effiziente Anpassung und Interaktion aller Microsoft-Anwendungen erreicht. Visual Basic erleichtert dem Programmierer durch seine grafische Oberfläche die Arbeit.

### **C und C++**

Die Programmiersprache C hat in der Praxis eine außerordentliche Bedeutung gewonnen. Durch den Aufschwung objektorientierter Sprachen wie C++ und Java, die auf C basieren, hat sich die Bedeutung von C noch erhöht. Entsprechend ihrer weiten Verbreitung wurde die Sprache standardisiert, zunächst durch das American National Standard Institute (ANSI) und inzwischen von der International Organization for Standardization (ISO). Programmierer, die sich an die normierten Konventionen halten, können ihre Programme relativ leicht auch auf andere Rechnertypen portieren.

C kennt Funktionen, aber keine Prozeduren. Die Stärke der Sprache C liegt in ihren vielen Operationen und ihrer Maschinennähe. Eine Schwäche ist die kaum stattfindende Überprüfung auf Typverträglichkeit, das kann zu Fehlern führen, die ein Programm zum Abstürzen bringen können, ohne dass im nachhinein die Ursache des Fehlers festzustellen ist. Die Weiterentwicklung von C ist C++.

C++ ist eine objektorientierte Sprache, das heißt, hier steht das Objekt im Vordergrund. Objekte, bestehen nicht nur aus sichtbaren Inhalten (Texten, Bildern, Grafiken...) sondern auch aus ergänzenden Verhaltensvorschriften und Bearbeitungsregeln. Mit objektorientierter Software kann beispielsweise eine Kalkulations-Tabelle in einen Text eingebaut werden. Alle tabellenkalkulations-typischen Eigenschaften der eingebauten Tabelle bleiben auch im Textdokument erhalten. Objekte sind weiterentwickelte (Zeichnungs-) Elemente. Man versucht, mit Mitteln einer Programmiersprache wie C++, Objekte wie Autos, Häuser, Firmen, Kunden, Rechnungen usw. nachzubilden. In der objektorientierten Programmierung wird das Objekt als unveränderlich betrachtet. Innerhalb eines Programms darf keine Funktion, die außerhalb des Projekts, zum Beispiel in einer Grafik oder einer Tabelle, beschrieben wurde, die bestehenden Attribute des Objekts verändern.

Die Programmiersprache C wurde 1972 von Dennis Ritchie an den AT&T Bell Laboratories im Zusammenhang mit dem Betriebssystem Unix entwickelt. Teile von C stammen aus der Programmiersprache BCPL. Bekannt wurde C durch das Buch "The

C Programming Language" im Jahre 1978. AT&T veröffentlichten das Unix-Betriebssystem an den ersten Universitäten. Ein kostenloser C-Compiler (Übersetzungsprogramm, das Programme einer höheren Programmiersprache, in Objektprogramme übersetzt), der K&R (Kerningham und Ritchie), wurde hinzugefügt. Jeder Unix-Betreiber konnte damit die Programmiersprache C nutzen. 1987 wurde der ANSI (American National Standard Institute)-Standard für C eingeführt. Mitte der achtziger Jahre entwickelte Stroustrup, an den AT&T Bell Laboratories C++. Die objektorientierte Programmierung wurde eingebunden und der C-Sprachumfang verbessert. Ursprünglich sollte "C++", "C mit Klassen" heißen. C++ steht für ein "verbessertes C". Mittlerweile gibt es C/C++ -Compiler für alle wichtigen Systeme. Die bekanntesten sind Turbo C/Borland C++ von Borland, sowie Microsoft C/ Visual C++ von Microsoft. Als nicht kommerzieller Compiler hat sich Gnu C etabliert.

### **Pascal, COBOL und andere**

Pascal, benannt nach dem französischen Mathematiker Blaise Pascal (1623 -1662) wurde 1972 von Niclas Wirth entwickelt. Ursprünglich war sie als eine Programmiersprache für Ausbildungszwecke gedacht. Sie hat sich aber inzwischen für eine Vielzahl von Anwendungen durchgesetzt, da sie einfacher als andere Programmiersprachen ist und die strukturierte Programmierung unterstützt. Pascal ist eine der höheren Programmiersprachen, die sich auch auf PCs einsetzen lassen. Durch eine leicht bedienbare Programmumgebung zeichnet sich die Weiterentwicklung Turbo-Pascal aus. Sie war auch eine der ersten auf dem PC verbreiteten Sprachen, die objektorientiertes Programmieren unterstützte. Mit Turbo-Pascal lassen sich Programme besonders schnell herstellen. Die amerikanische Softwarefirma Borland entwickelte, durch den großen Erfolg von Visual Basic veranlasst, eine entsprechende Entwicklungsumgebung auf der Basis von Pascal und nannte sie Delphi. Delphi ist ein Compiler, der den objektorientierten Pascal-Code in Maschinensprache übersetzt. COBOL (Common Business Oriented Language) ist eine Programmiersprache, die zwischen 1959 und 1961 entwickelt wurde. Zunächst wurde sie im US-Verteidigungsministerium eingesetzt, später für Anwendungen im geschäftlichen Bereich auf Großrechnern. Diese problemorientierte Programmiersprache wurde stets gepflegt und fortgeführt, die Grundstruktur allerdings beibehalten. Programme, die mit COBOL entwickelt sind, bauen auf vier Elementen (Divisions) auf. Sie müssen immer in fester Reihenfolge auftreten. Mit dem Erkennungsteil (Identification Division) wird der Programmname festgelegt, mit dem man es in der Programmbibliothek findet, ebenso Erläuterndes, Name des Programmierers, Kommentare und anderes. Im Maschinenteil (Environment Division) wird der einzusetzende Computer sowie die für Ein- und Ausgabe wichtigen Dateien festgelegt und verwaltet (Rechner und Peripheriegeräte). Im Datenteil (Data Division) findet man die Beschreibung und Festlegung der Datenstruktur (Daten, Dateien, Bereiche im Arbeitsspeicher), im Prozedurteil (Procedure Division) liegt der eigentliche Ablauf des Programms, der letztendlich die Aktionen des Programms festlegt.