

Einführung in die Informatik, Algorithmen und Datenstrukturen

Teil 1 - Thema 4
Programmiersprachen

Programmiersprachen – Angebotsformen, Klassifikation und Entwicklung

Programmiersprachen

„Was ist Programmieren? Manche sagen, es sei eine Kunst, andere, eine Wissenschaft, wieder andere, ein Handwerk. Ich sage: keines alleine, aber von jedem etwas.“

Ada Lovelace (1815-1852)

Didaktische Auswahl der Programmiersprachen

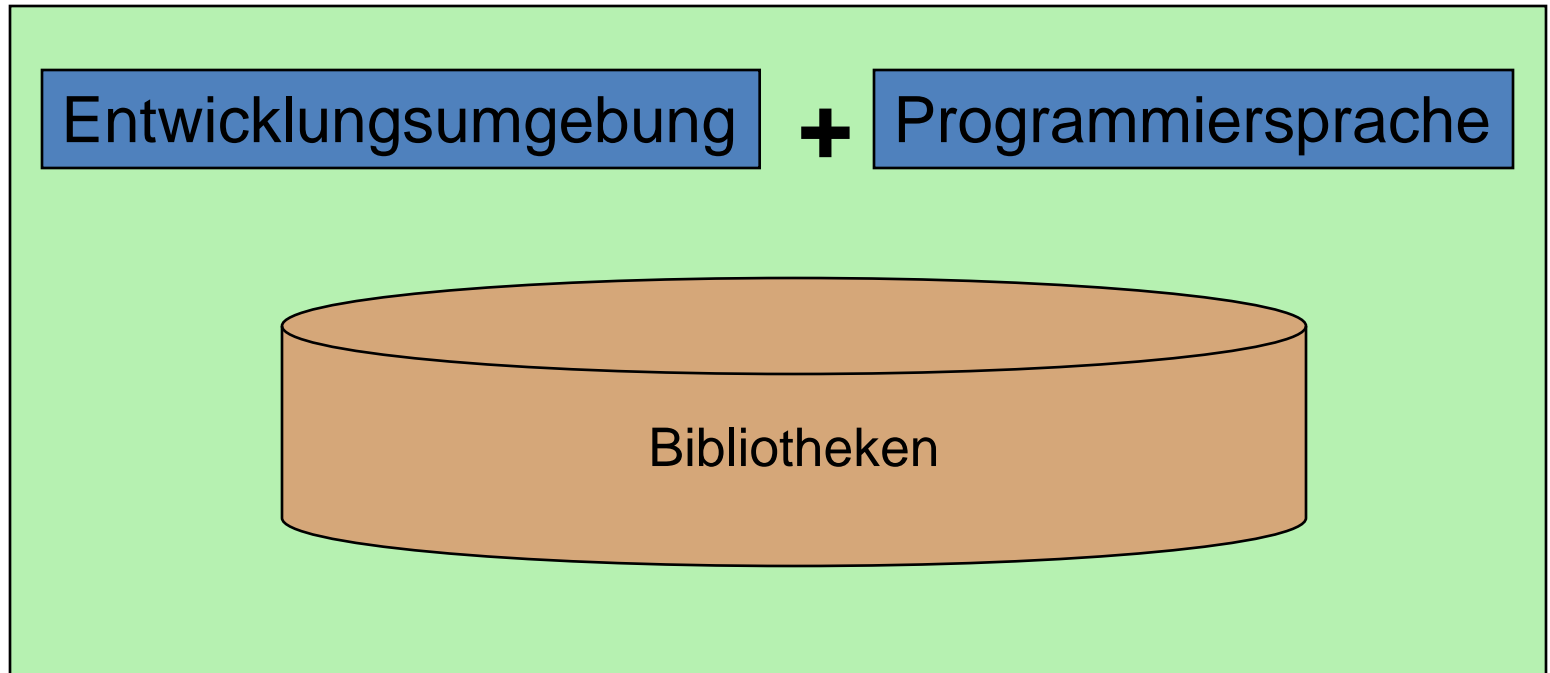
Professionelle Softwareentwicklungssysteme

- komplexe Systeme
- auf kommerzielle Softwareentwicklung ausgerichtet
- weit verbreitet
- Ausbildung an einem System, welches in der weiteren Ausbildung bzw. in der beruflichen Tätigkeit eingesetzt werden kann

Spezielle Ausbildungssprachen

- kleine Systeme
- didaktisch auf die Vermittlung der algorithmischen Grundstrukturen und informatischen Denkweisen optimiert
- einheitliches Ausgangsniveau
- für den praktischen Einsatz müssen die erlernten Techniken und Methoden auf ein anderes Entwicklungssystem übertragen werden

Programmierwerkzeuge



Anforderungen an ein Programmierwerkzeug

- **unterschiedliche Modellierungsarten** sollen mit einem Werkzeug realisierbar sein
- **klarer und intuitiver Sprachaufbau** (Syntaxelemente sollen die damit verbundene Semantik deutlich machen)
- **klares, ausgereiftes Typenkonzept**
- **präzise Fehlermeldungen** und geeignete Debugginghilfsmittel
- **geeignetes Hilfesystem**
- **unterstützendes Entwicklungssystem**

Angebotsformen von Programmierwerkzeugen

Kommerzielle Programme

Kommerzielle Programme werden als Lizenzen über den Handel vertrieben. Beim Kauf des Programms wird in der Regel ein Nutzungsvertrag geschlossen, in dem sich der Käufer verpflichtet, die Nutzungsbedingungen des Herstellers zu akzeptieren.

Lizenzen können als Einzelplatzlizenzen, als Netzlizenzen ggf. mit beschränkter Platzzahl, als Schullizenzen oder auch als Landeslizenzen erworben werden.

Angebotsformen von Programmierwerkzeugen

Updates

Updates (Aktualisierungen) sind Programme, die als Folgeversion für bereits vorhandene Lizenzen erworben werden können. Sie sind in der Regel preiswerter als eine neue Lizenz bzw. kostenlos.

Von einigen Herstellern werden jedoch für Schullizenzen keine Updates angeboten, so dass jeweils eine neue Schullizenz zu erwerben ist.

Angebotsformen von Programmierwerkzeugen

Shareware

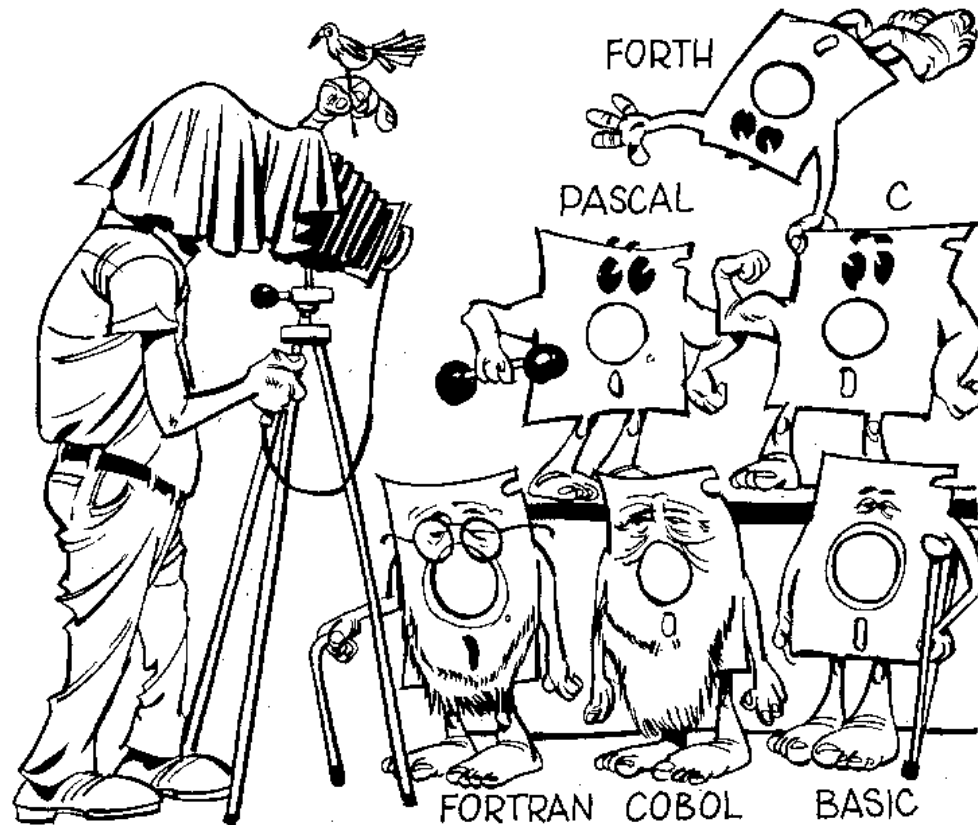
Als Shareware-Programm wird Software bezeichnet, die **kostenlos getestet und weitergegeben** werden kann. Sollen diese Programme jedoch dauerhaft genutzt werden, so ist es notwendig, sich registrieren zu lassen und eine entsprechende Registrierungsgebühr zu entrichten. Dann erhält man in der Regel das Handbuch und ggf. eine aktuellere Version. Ein Herstellersupport wird von den Shareware-Entwicklern kaum angeboten.

Angebotsformen von Programmierwerkzeugen

Freeware

Freeware wird teilweise auch als Public-Domain bezeichnet. Dies kann von jedermann **kostenlos genutzt werden**. Neben einer Vielzahl von minderwertigen Produkten stehen aber im Freeware-Bereich auch anspruchsvolle Produkte zur Verfügung. Beispiele dafür sind die Satzprogramme TeX und LaTeX, Postscript-Ausgabeprogramme (Ghostscript), Programmiersprachen (GNU-C-Compiler) oder das Betriebssystem Linux. Freeware darf ohne Einschränkungen verändert, kopiert oder weitergegeben werden. Sie muss im Quellcode vorliegen. Es gibt keine Copyrights an dieser Software (international).

Programmiersprachen



Quelle: Peter Norton, MS-DOS und PC-DOS, 1985

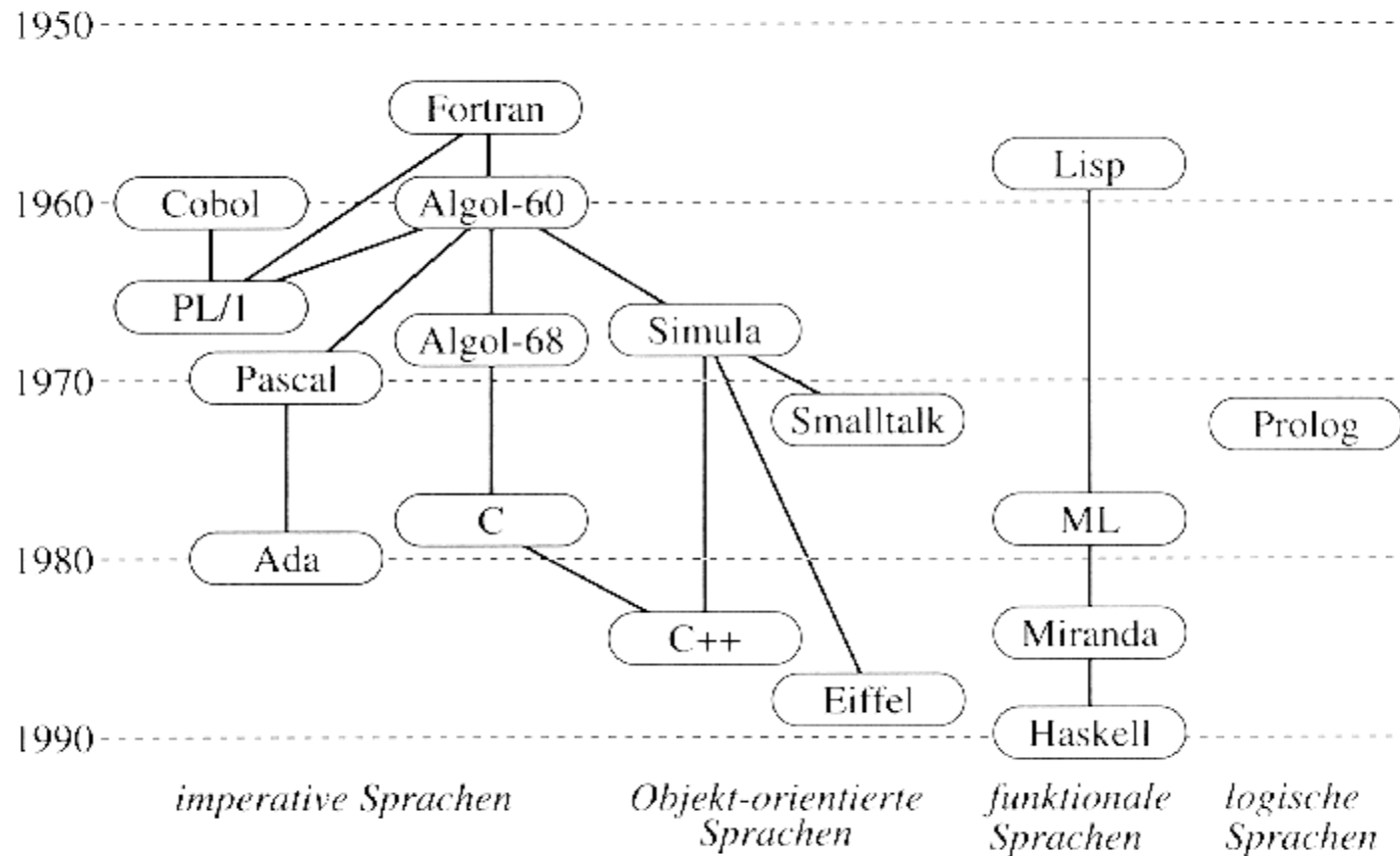
Geschichte der Programmiersprachen

- Entwicklung der **ersten höhere Programmiersprache** (für Computer), **Plankalkül**, von Konrad Zuse 1942-46
- umfasst unter anderem Wertzuweisungen, Unterprogramme, bedingte Anweisungen, Schleifen, eine Gleitkommaarithmetik, Feldvariablen, zusammengesetzte Datentypen
- 1972 erstmals veröffentlicht,
- 1998 erstmals implementiert
- 2000 an der FU Berlin implementiert

Sprachniveaus von Programmiersprachen

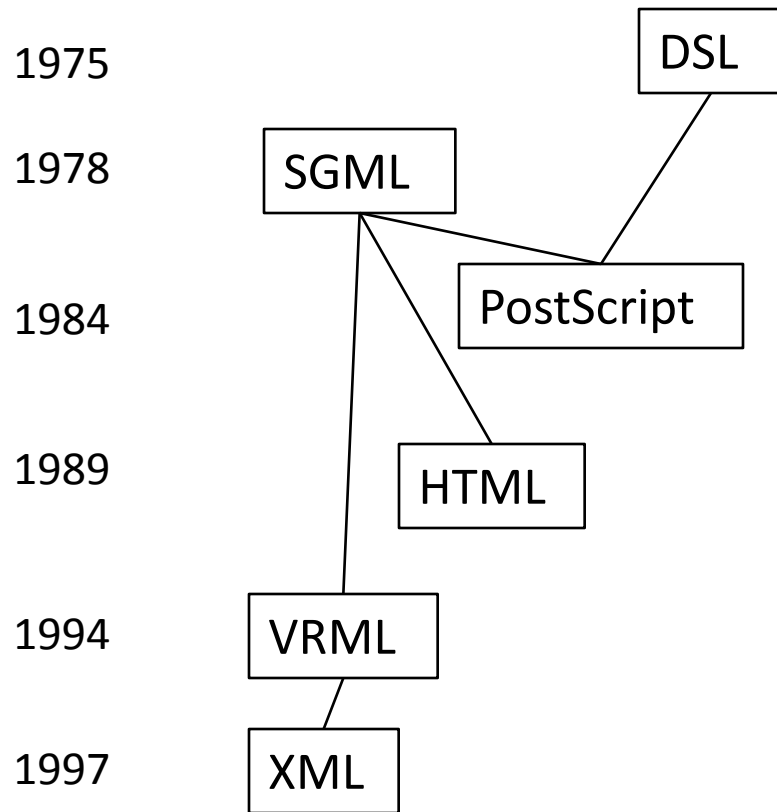
NUTZER
anwendungsorientierte Sprachen
verfahrensorientierte Sprachen
Makrosprachen
Assemblersprachen
Maschinensprachen
Mikroprogrammiersprachen
GERÄTETECHNIK

Entwicklungslinien von Programmiersprachen (Auswahl)



Quelle: David A. Watt, Programmiersprachen, 1996

Deskriptive Sprachen



Sprachen der 1. Generation - Maschinensprachen

Maschinensprachen

- Imperatives Paradigma
- Befehle werden binär codiert
- Befehle werden direkt von der Hardware bzw. CPU abgearbeitet
- es werden prozessorspezifische Codes, meist in Verbindung mit Daten an den Prozessor gesandt
- Befehle verschieben in der Regel Daten zwischen Speicher und Register und veranlassen die Ausführung einfacher arithmetischer Operationen

Eine **Maschinensprache** ist eine Programmiersprache, deren Alphabet nur aus zwei Zeichen besteht und vom Computer direkt verarbeitet werden kann.

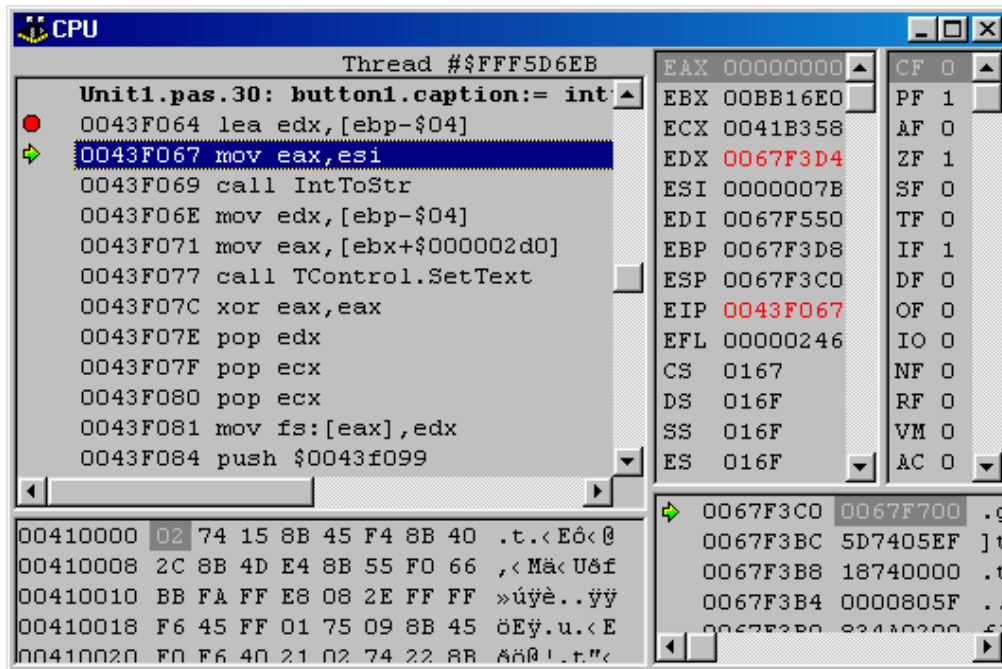
© Bibliographisches Institut & F.A. Brockhaus AG Mannheim und paetec Gesellschaft für Bildung und Technik mbH

Sprachen der 2. Generation - Assemblersprachen

Assemblersprachen

- Imperatives Paradigma, noch maschinenorientiert (registerorientiert)
- Statt binärer Codierung für Befehle werden mnemonische Abkürzungen benutzt, statt Adressen für Speicherzellen werden Bezeichnungen für Variable verwendet.
- repräsentiert die Maschinensprache einer spezifischen Prozessorarchitektur in einer für den Menschen lesbaren Form
- jede Computerarchitektur hat ihre eigene Assemblersprache

Delphi-Debugger



```
procedure TForm1.Button1Click(Sender: TObject);  
  var i : integer;  
begin  
  i:= 123;  
  button1.caption:= inttostr(i);  
end;
```

Sprachen der 3. Generation - höhere imperative Programmiersprachen

Das **Problem und seine Lösung** in Schritten (Algorithmus) **stehen im Vordergrund**. Abstraktion von der Maschinenorientierung , Hinwendung zur anwendungs- und bedienerfreundlichen Problemorientierung.

Es entstanden

- die prozeduralen Sprachen (FORTRAN ab 1954)
- die objektorientierten Sprachen (Smalltalk ab 1970)
- die hybriden Sprachen (C++, Turbo Pascal ab 5.5)
- die domänenspezifischen Sprachen
- die deklarativen Sprachen (LISP ab 1960)

FORTRAN

1954 bis 1956: FORTRAN (FORmular TRANslator):

Anwendungsgebiet: technisch-wissenschaftliche Berechnungen
(numerische Aufgaben)

Sprachklasse: prozedural (imperativ) mit objektorientierten Komponenten

Zu den wesentliche Eigenschaften, die mit FORTRAN in die Programmierung eingeführt wurden, gehörten

- Felder
- Schleifen, die durch Indizes gesteuert werden konnten und
- die verzweigende IF-Anweisung.

Aktuelle Version: FORTRAN 2003 mit folgenden Eigenschaften:

- Die Quellprogramme können formatfrei geschrieben werden.
- Es besteht die Möglichkeit, die Genauigkeit von Variablen zu spezifizieren.
- Bit - Operationen sind möglich.
- Es sind moderne Schleifenstrukturen und die CASE-Anweisung implementiert.

COBOL

1959-1960: COBOL (COmmon Busines Oriented Language)

Anwendungsgebiet: Verarbeitung großer Datenmengen (Massendatenverarbeitung), Belegverarbeitung, Finanzbuchhaltung

Die wesentlichen Eigenschaften, die COBOL zum Sprachentwurf beigegeben hat, sind:

- die Trennung der Datenstrukturen vom Ausführungsteil eines Programms
- vielseitige Formatierungsmöglichkeiten für die Ausgabe unter Verwendung von „Pictures“
- die RECORD-Struktur zur Organisation von Daten

ALGOL

1958 bis 1960: ALGOL (ALGOrithmic Language)

Anwendungsgebiet: Beschreibung und Programmierung vorrangig von Algorithmen der numerischen Mathematik

Sprachklasse: prozedural (imperativ)

Für die Sprachentwicklung wurden mit ALGOL folgende Konzepte eingeführt:

- Formatfreiheit,
- strukturierte Anweisungen,
- begin-end-Blöcke,
- Typdeklarationen für Variable,
- Rekursionen und
- call-by-value-Parameter (Wertparameter).

ALGOL war die erste Sprache, die die Notation der Backus-Naur-Form benutzte, um die Syntax zu definieren.

LISP

LISP (LISt Processor - 1958)

Anwendungsgebiet: Anwendungen in der Künstlichen Intelligenz

Sprachklasse: symbolorientierte Sprache, funktionale Sprachen

LISP wurde auf der Grundlage allgemeiner Listenstrukturen und Funktionsanwendungen von John Mc Carthy am MIT entworfen.

Aktuelle Versionen sind Common Lisp und Scheme.

LISP basiert auf einer einheitlichen Datenstruktur, auf dem S-Ausdruck und der Funktionsanwendung. Mit der Einführung des Prinzips der „Garbage Collection“ wurde das Verfahren der Laufzeit-Speicherplatzzuteilung realisiert.

Gehört zur Generation Sprachen, die eine deklarativ-funktionale Beschreibung von Problemen ermöglichen.

SIMULA 67

1965-1967: Simula67

Anwendungsgebiet: Simulationssprache für Prozesssimulation

Sprachklasse: domänenspezifische Sprachen

Sie basiert auf Simula I und enthält ALGOL60 als Teilmenge.

Mit der Einführung des Klassenkonzepts wurde diese Sprache zur **ersten objektorientierten Programmiersprache**.

BASIC

BASIC (Beginners All-purpose Symbolic Instruction Code -1963)

Anwendungsgebiet: Ausbildungssprache, Makrosprache in Applikationen

Sprachklasse: prozedural (imperativ), mit objektorientierten Elementen

Nachteile:

- unstrukturiert, da zeilenorientiert,
- keine Blockanweisungen,
- kein strenges Typenkonzept und
- häufig nur als Interpretersprache verfügbar

Aktuelle Versionen:

- VBA (Visual BASIC for Applications) in Office-Anwendungen und
- VB.NET (Visual Basic.NET)

Pascal

1971: PASCAL (1983: Turbo-Pascal)

Anwendungsgebiet: Ausbildungssprache

Sprachklasse: prozedural (imperativ), objektorientiert (ab Version 5.5)

Kennzeichen:

- einfache, strukturierte Sprache
- strenge Typisierung
- klares Unterprogrammkonzept
- Modularisierung
- dynamische Speicherverwaltung

letzte kommerzielle Versionen: Borland-Pascal 7 (wird nicht mehr angeboten)

Delphi/Object-Pascal

1995 Delphi/Objekt-Pascal

Anwendungsgebiet: Erstellung von Windows-Applikationen

Sprachklasse: prozedural (imperativ), objektorientiert

Eigenschaften:

- objektorientiert,
- ereignisgesteuert und
- visuell.

Über die Together-UML-Schnittstelle wird die Entwicklung objektorientierter Applikationen unterstützt.

Aktuelle Version: Code Gear RAD-Studio 2009 – Delphi für Win 32, (KYLIX Vers. 3 für UNIX-Anwendungen), Turbo-Delphi

Imperative und objektorientierte Programmiersprachen – C/C++/C#

C (1972), C++ (1980), C# (2001)

Anwendungsgebiet: Systemprogrammiersprache, Echtzeitprogrammierung

Sprachklasse: prozedural (imperativ), objektorientiert (ab C++)

Die Programmiersprache C wurde in den Bell Laboratories von Dennis Ritchie entwickelt.

Die Sprache ist blockorientiert und erlaubt eine sehr kompakte Schreibweise. Ein C-Programm ist eine Menge von Funktionen, die einander aufrufen können.

Es wurde in den frühen achtziger Jahren zur Version C++ weiterentwickelt. Damit steht ein Werkzeug für die Entwicklung großer Softwaresysteme zur Verfügung. **C++ besitzt leistungsfähige Komponenten zur objektorientierten Programmierung.**

PROLOG

Bei logischen Sprachen ist ein Programm die Niederschrift von Fakten ("Prädikaten", Definitionen) und Regeln, womit der Computer neue Fakten gewinnt und das Problem gelöst wird. Ein Vertreter ist beispielsweise PROLOG.

© Bibliographisches Institut & F.A. Brockhaus AG Mannheim und paetec Gesellschaft für Bildung und Technik mbH Berlin. Alle Rechte vorbehalten. www.schuelerlexikon.de

PROLOG ist eine Programmiersprache, die auf die Lösung von Problemen mit Hilfe des Prädikatenkalküls ausgerichtet ist.

Anwendungsbereiche: KI-Systeme, Datenbankabfragen, mathematische Beweise

Sprachklasse: logische Programmiersprache

Für Prolog existiert kein Standard, jedoch wird die an der Universität Edinburgh entwickelte Version heute am häufigsten verwendet.

Prolog wird in der Regel unter Kontrolle eines Interpreters ausgeführt.

SMALLTALK

1972 bis 1980: Smalltalk

Anwendungsgebiet: Verwaltungssysteme,
Massendatenverarbeitung

Sprachklasse: objektorientierte Programmiersprache

Smalltalk wurde so entworfen, dass es den objektorientierten Ansatz auf eine vollständig konsistente Weise anwendet. Besonders für Mainframe-Applikationen nimmt die Bedeutung von Smalltalk immer mehr zu, da vielfach veraltete COBOL-Anwendungen durch Smalltalk-Applikationen ersetzt werden. Grundlage für Entwicklungssysteme: Squeak, EToys

Java

1991-92 Java (1995)

Anwendungsgebiet: plattformunabhängige Sprache in Netzanwendungen

Sprachklasse: objektorientierte Programmiersprache

Java ist eine einfache, konsequent objektorientierte, multithreaded, robuste, architekturneutrale, übertragbare und dynamische Programmiersprache.

Sie basiert auf der Programmiersprache C++, wurde aber wesentlich vereinfacht. Java-Programme werden in Bytecode kompiliert und können damit ohne weitere Kompilierung auf verschiedenen Computern ausgeführt werden, sofern es eine Java Virtual Machine dafür gibt.

Professionelle Entwicklungsumgebungen stehen mit dem „Java Development Kit“ von Sun und Visual J++ von Microsoft zur Verfügung.

JavaScript

Java Script (1995)

Anwendungsgebiet: überwiegend clientseitige Programmierung von Webseiten

Sprachklasse: objektbasiert, prozedural

- Einbinden in HTML-Seiten möglich
- Datenein- und -ausgabe über Formulare
- keine strenge Typisierung
- interpretative Abarbeitung

Literatur:

- /DUDEN03/ Basiswissen Schule – Informatik Abitur
PAETEC Verlag für Bildungsmedien Berlin, 2003
ISBN 3-89818-065-4
- /HENNING07/ Henning, Peter A., Holger Vogelsang
Taschenbuch Programmiersprachen
Fachbuchverlag Leipzig, 2007;
ISBN 978-3-446-40744-2
- /HORN03/ Horn, Christian; Immo O. Kerner; Forbig, Peter
Lehr- und Übungsbuch Informatik
Band 1: Grundlagen und Überblick
Fachbuchverlag Leipzig; 2003; ISBN 3-446-22543-9
- /FORBIG06/ Forbig, Peter; Immo O. Kerner
Lehr- und Übungsbuch Informatik
Programmierung – Paradigmen und Konzepte
Fachbuchverlag Leipzig; 2006; ISBN 3-446-40301-9
- /LOUDEN94/ Louden, Kenneth C.
Programmiersprachen
Thomson Publishing International
Bonn, 1994; ISBN 3-929821-03-6