

Programmiersprachen

Einführung

Inhalt

Wozu — weshalb — warum?
 tabellarische Geschichte der Programmiersprachen
 Syntax — Konzept — Semantik — Methodik
 Was ist ein Konzept? Ein Abriss anhand der *Zählschleife*

Anmerkung

dies ist eine kurze Einführung mit Vorlesungs- und Übungs-Anteilen

Was erwartet ihr von einer Programmiersprache?

erwünschte Eigenschaften

nicht erwünschte Eigenschaften

Wozu brauche ich *eine* Programmiersprache?

Zum Programmieren natürlich!

Aber wer muss eigentlich noch programmieren?

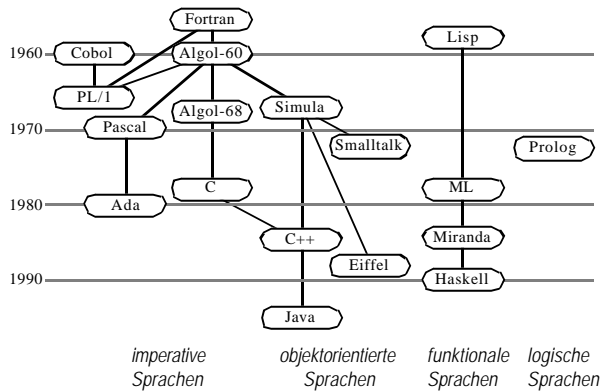
Wozu braucht es *mehrere* Programmiersprachen?

für verschiedene Anwendungsgebiete (kommerziell, KI, verteilte Systeme, Internet ...)
 für verschiedene Plattformen (Windows, Unix, Mainframes, Vektorrechner ...)
 für verschiedene Anforderungen (sicherheitskritisch, eingebettet, interaktiv, effizient, kostengünstig, methodisch sauber)

Welche Programmiersprachen kennt Ihr?

Kenntnis Sprache	Kenntnis			Kenntnis Sprache	Kenntnis			Kenntnis Sprache	Kenntnis		
	flüchtig	gut	sehr gut		flüchtig	gut	sehr gut		flüchtig	gut	sehr gut
Ada				Lisp				Scheme			
Algol				Miranda				Smalltalk			
Basic				ML				Simula			
C				Modula-2							
C++				Modula-3							
Cobol				Oberon							
Eiffel				occam							
Fortran				Pascal							
Haskell				PL/I							
Java				Prolog							

tabellarische Geschichte der Programmiersprachen



objektorientierte Sprachen in Stichworten

ab 1967

Simula: erweiterbare Strukturen, Klassen, Untertypen
 Smalltalk: Zugriffsschutz, Nebenläufigkeit, Speicherbereinigung
 C++: Mehrfachvererbung, generische Klassen
 Eiffel: Mehrfachvererbung, generische Klassen, Zusicherungen
 Oberon: ?
 Java: Plattformunabhängigkeit

funktionale / logische Sprachen in Stichworten

funktional (ab 1956)

Lisp: Listen, Speicherbereinigung, Rekursion
 ML: *pattern matching* Funktionale, Polymorphie und Module
 Miranda: verzögerte Auswertung
 Haskell: Typklassen, Monaden

logisch (ab 1972)

Prolog: Variablen, Resolution, *backtracking*

Was macht eine Programmiersprache aus?

"künstliche" Computersprache

implementierbar

eindeutige Beschreibung von Berechnungen (Algorithmen)

Universalität (im Unterschied zu anderen Computersprachen)

geeignet zur Beschreibung von beliebigen Berechnungen (allen beschreibbaren)

Natürlichkeit

getreue Modellierung von Systemen

höhere Programmiersprachen

sollten *plattformunabhängig* sein