Programmiersprachen

Organisation und Einführung

Berthold Hoffmann

Studiengang Informatik
Universität Bremen

Sommersemester 2010
(Vorlesung am Montag, der 12. April 2010)

(Montag, der 12. April 2008)

Gestatten?

Berthold Hoffmann
wissenschaftlicher Mitarbeiter
Arbeitsgruppe Krieg-Brückner

Kordinaten

- Büro: Cartesium 2.48
- Telefon: 218-64 222
- Email: hof@inf...men.de
- `www.inf...men.de/~hof`

Interessen

- Übersetzer
- Programmiersprachen
- visuelle Sprachen
- Graphtransformation
- Dia**plan**

Der Inhalt in der Nußschale

Prinzipien von Programmiersprachen

- Was ist allen Sprachen gemeinsam?
  $\Rightarrow$ Konzepte und ihre Eigenschaften
- Was prägt den Stil einer Sprache?
  $\Rightarrow$ Paradigmen
- Was macht guten Sprachentwurf aus?
  $\Rightarrow$ Prinzipien
- Aspekte von [Programmier-] Sprachen
  - Syntax: die äußere Form (wie?)
  - Semantik: die Bedeutung (was?)
  - Pragmatik: der Zweck (wozu?)

Lehrziele / Lernziele

Was können Sie (kennen) lernen?

- andere Sprachen, andere Sprachkulturen
- Erlernen und Vergleichen von Sprachen
- Entwurf von Sprachen

Nicht-Lehrziele / Nicht-Lernziele

- Programmiermethodik
- Programme in einer Sprache X entwickeln
- Aber: viele Konzepte und Stile sind methodisch motiviert

Inhalt

Einführung

- Geschichte

Konzepte

- Werte
- Speicher
- Bindung
- Abstraktion
- Kapselung
- Typsystem
- Steuerung

Paradigmen

- imperativ (Ada, C)
- objektorientiert (Eiffel, Java)
- funktional (ML, Haskell)
- logisch (Prolog)
- Skriptsprachen (?)

Spachentwurf

- Entwurfsprinzipien
- Sprachauswahl

Lehrziele

Konzepte

- Werte – Datentypen, Ausdrücke, Typisierung
- Speicher – Variablen, Zeiger, Befehle
- Bindung – Vereinbarungen, Blöcke
- Abstraktion – Prozeduren, Parameter,
- Kapselung – Module, Klassen, generische Module
- Typsystem – Überladen, Polymorphie, Vererbung
- Steuerung – Sprünge, Auswege, Ausnahmen

(Montag, der 12. April 2008)

Material zur Veranstaltung

www.inf...men.de/agbkb/lehre/programmiersprachen/

- Lehrbuch zur Veranstaltung: David A Watt
  Programmiersprachen – Konzepte und Paradigmen.
  München: Hanser, 1996. (Deutsch von B. Hoffmann)
- Folienkopien
- Hinweise zu Programmiersprachen
- Aufgaben
- Hintergründe

(Montag, der 12. April 2008)

Wozu brauchen wir Programmiersprachen?

- Zum Programmieren natürlich
  Aber wer muss eigentlich noch programmieren?
- Und weshalb reicht nicht eine einzige Sprache?
  - verschiedene Anwendungen
    (kommerziell, KI, verteilte Systeme, Internet ...)
  - verschiedene Plattformen
    (Windows, Unix, Mainframes, Vektorrechner ...)
  - verschiedene Anforderungen
    (sicherheitskritisch, eingebettet, interaktiv, effizient, kostengünstig, wiederverwendbar)

Was macht eine Programmiersprache aus?

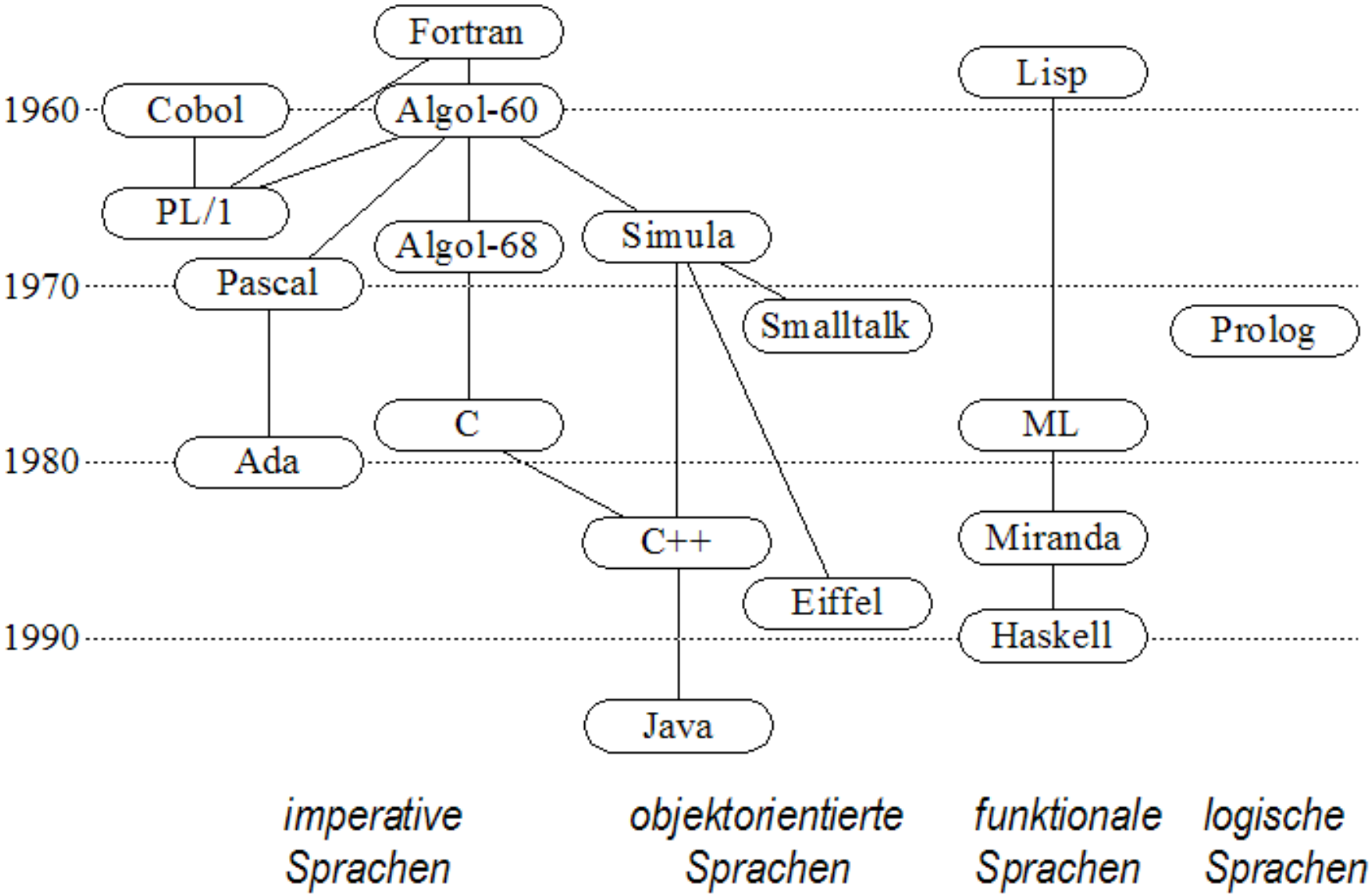
- Computersprachen / formale Sprachen
  - implementierbar
- für eindeutige Beschreibung von Algorithmen
  - universell:
    alle berechenbaren Funktionen müssen beschreibbar sein
- “höhere” Sprachen
  - Plattformunabhängigkeit
- Pragmatik
  - natürlich
  - effizient

Was erwarten wir von Programmiersprachen?

- erwünschte Eigenschaften

- unerwünschte Eigenschaften

Geschichte der Programmiersprachen



imperative Sprachen in Stichworten

ab 1957

Fortran	Ausdrücke, Felder und Unterprogramme
Cobol	Strukturen und Dateien
Algol-60	Blöcke und rekursive Prozeduren
PL/1	Datentypen, Ausnahmen und Nebenläufigkeit (*low level*)
Algol-68	Datenstrukturen, Kontrollstrukturen, Nebenläufigkeit (*low level*)
Pascal	Kontrollstrukturen, Datenstrukturen
C	?
Ada	(parametrisierte) Module, Ausnahmen und Nebenläufigkeit

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
C#	.NET, Interoperativität zwischen Sprachen

Funktionale / logische Sprachen in Stichworten

funktional (ab 1956)

- Lisp: Listen, Speicherbereinigung, Rekursion
- Ml: pattern matching , Funktionale, Polymorphie und Module
- Miranda: verzögerte Auswertung, Listenumschreibungen
- Haskell: Typklassen, Monaden

logisch (ab 1972)

- Prolog: Variablen, Resolution, backtracking

Kombinationen funktionaler und logischer Sprachen

- Curry: Haskell plus Prolog
- Oz: objektorientiert, funktional, logisch, nebenläufig

(Montag, der 12. April 2008)

Zusammenfassung

- Inhalt: Prinzipien von Programmiersprachen
- Konzepte – Paradigmen – Entwurfsprinzipien
- Syntax – Semantik – Pragmatik
- kaum Methodik – kaum Implementierung
  ($\Rightarrow$ Übersetzer, Sommer 2009)
- Referenzsprachen Ada – Eiffel – Ml – Prolog

Nächstes Mal

- Werte
  - Datentypen und Datenstrukturen
  - Ausdrücke
  - Typisierung

Weitere Lehrbücher

beamericonbook David A. Watt.
Programming Language Design Concepts.
Chichester: Wiley & Sons, 2004

beamericonbook Robert W. Sebesta.
Concepts of Programing Languages, 5/e.
Hemel Hampstead: Addison-Wesley, 2003

beamericonbook Es geht aber auch ohne Buch!
(bzw. mit der Übersetzung von D.A. Watt's Buch im Netz)