

6. Programmiersprachen im Überblick

Was ist das: eine Programmiersprache?

- Formale Sprache^{*)}, die zur Beschreibung von Berechnungen in Computern verwendet wird:

Programme = Daten + Algorithmen

^{*)} Mengen von Wörtern über einem Alphabet: $L \subseteq A^*$
(Folge von Symbolen)

- Wichtig:
 - Sprache für menschlichen Leser verständlich;
 - effizient implementierbar
- Sprache hat Syntax (vgl. Grammatiken, EBNF) und Semantik (Bedeutung, Wirkung).

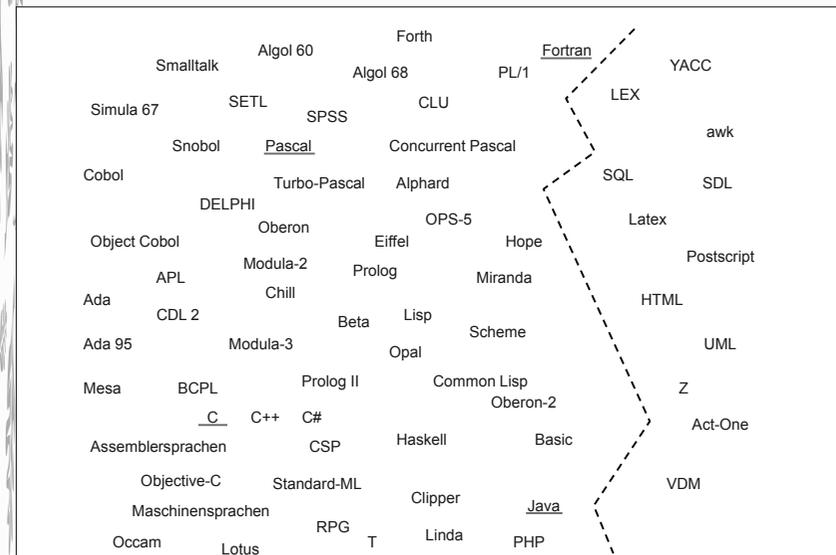
Probleme und Fragen

Mehr als 1000 unterschiedliche Programmiersprachen in Forschungsgruppen, internationalen Komitees sowie in Computerfirmen entwickelt [Wilson 93]

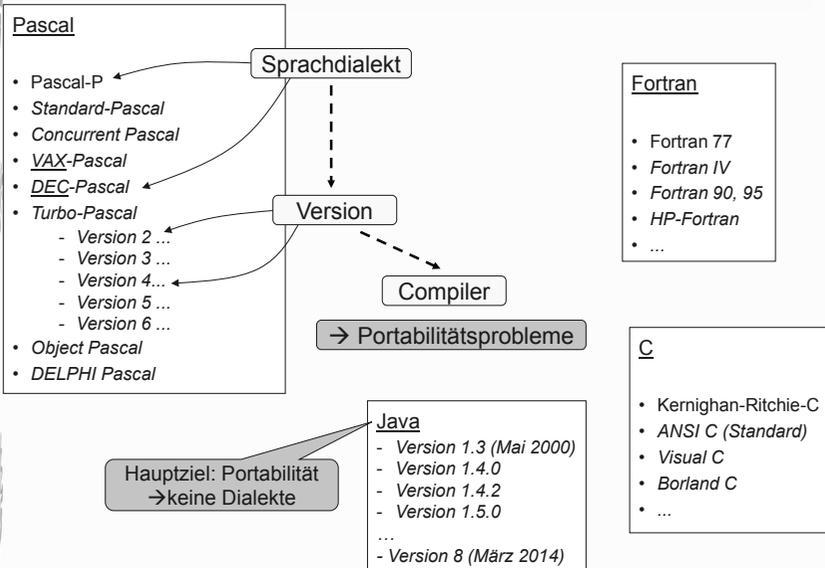
IBM: Fortran, PL1
Sun: Java
IFIP: Algol 60, Algol 68
Universitäten: Pascal, Modula-2, CLU
DoD: Cobol, Ada

- Sprachen, Dialekte, Versionen
... und das Hauptproblem der Vielfalt: Portabilität
- Klassifikation:
Wie kann eine Ordnung in die Vielfalt gebracht werden?
- Klassifikation ...
 - ... nach Anwendungsgebieten
 - ... nach der Historie
 - ... nach Programmiersprachengenerationen
 - ... nach Programmierparadigmen
 - ... nach Verbreitungsgrad

Programmiersprachen: eine Auswahl



Programmiersprachen und Varianten



Welche Programmiersprache ...

- ... sollte man beherrschen?
- ... ist weit verbreitet?
- ... ist die beste?
- ... ist modern?

Sollte man möglichst viele Programmiersprachen beherrschen?

Wie kann eine Ordnung in die Vielfalt von Programmiersprachen gebracht werden?

Klassifikationen

- ... nach Anwendungsgebieten
- ... nach der Historie
- ... nach Programmiersprachengenerationen
- ... nach Programmierparadigmen
- ... nach Verbreitungsgrad

Klassifikationen

- ... nach Anwendungsgebieten

Programmiersprachen und Anwendungen (1)

- *Wissenschaftlich-technischer Bereich (Physik):*
Fortran
- *Kommerzieller Bereich (Banken, Verwaltung):*
Cobol
- *Künstliche Intelligenz (Expertensysteme ...):*
Prolog, Lisp, Haskell ...
- *Systemsoftware (Compiler, Betriebssysteme):*
C, C++, Ada, Java, CDL 2
- *Echtzeit:*
Ada95, Pearl
- *Datenbanken:*
SQL
- *Telekommunikation:*
Chill

Achtung: Viele Sprachen sind universell anwendbar (C, C++, Ada, Java, Pascal, Python ...)

Programmiersprachen und Anwendungen (2)

- *Statistik:*
SPSS
R (statistische Auswertung von „big data“, 1990, populär ab 2010)
- *Schriften / Texte:*
Latex, Postscript
- *Compiler-Generatoren:*
Lex, Yacc
- *formale Spezifikationen:*
Z
- *SW-Architekturen:*
UML

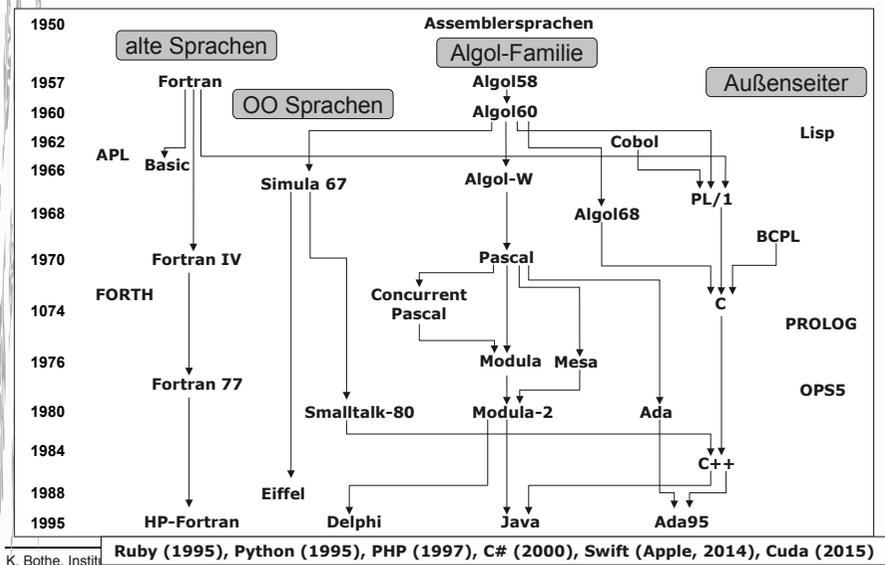
Nicht mehr Programmiersprachen im engeren Sinne

Klassifikationen

... nach der Historie

Entwicklung von Programmiersprachen

(Quelle: nach Appelrath, Ludewig 'Scriptum Informatik', 1995)



Internet: Informationen zu Programmiersprachen

- ▶ **Liste (aller) Programmiersprachen** (Stand 6. 10. 2015):
http://de.wikipedia.org/wiki/Liste_von_Programmiersprachen
- ▶ **Liste objektorientierter Programmiersprachen** (Stand 30. Mai 2015):
http://de.wikipedia.org/wiki/Liste_objektorientierter_Programmiersprachen
- ▶ **Zeittafel:**
http://de.wikipedia.org/wiki/Zeittafel_der_Programmiersprachen

Klassifikationen

... nach Programmiersprachengenerationen

Programmiersprachengenerationen

- Programmiersprachen der 1. Generation:
Maschinensprachen
- Programmiersprachen der 2. Generation:
Assemblersprachen
- Programmiersprachen der 3. Generation:
höhere algorithmische u. objektorientierte Sprachen
→ Pascal, Ada, C, Basic, Java ...
- Programmiersprachen der 4. Generation:
Tabellenkalkulation, Datenbanksprachen
→ Lotus, SuperCalc, dBase ...
- Programmiersprachen der 5. Generation:
Sprachen der künstlichen Intelligenz
→ Prolog, Lisp, OPS-5 ...
deskriptive Sprachen

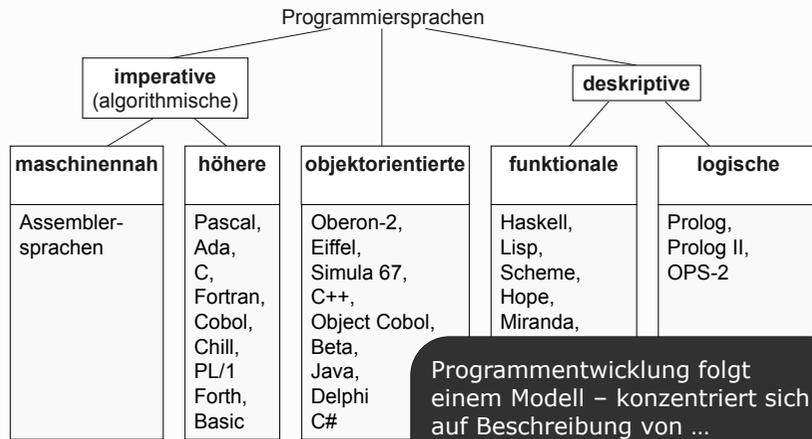
→ Abstraktionsniveau

Klassifikationen

... nach Programmierparadigmen

Ganz wichtig!

Programmiermodell (Paradigma)



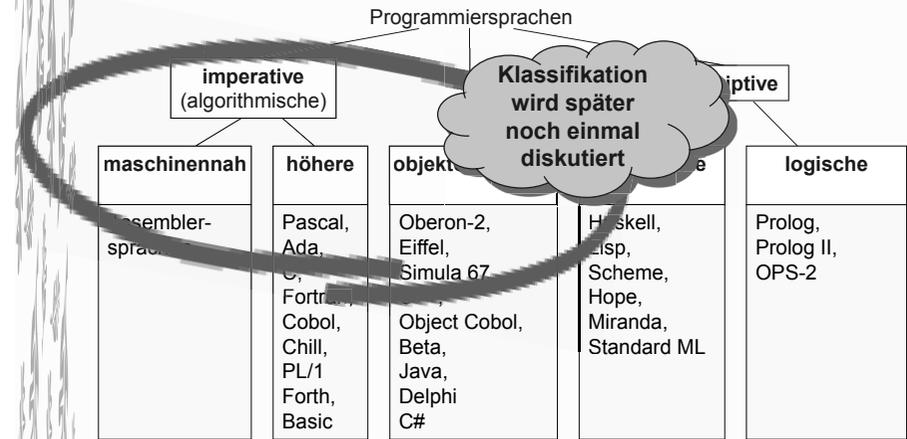
Programmentwicklung folgt einem Modell – konzentriert sich auf Beschreibung von ...

- Algorithmen
- Objekten
- Funktionalen Zusammenhängen
- Logik des Problems

Achtung: Mischformen

- Java: objektorientiert, imperativ
- Common Lisp: objektorientiert, funktional, imperativ

Programmiermodell (Paradigma)



Achtung: Mischformen

- Java: objektorientiert, imperativ
- Common Lisp: objektorientiert, funktional, imperativ

Programmiermodell: imperativ

- Orientiert am Algorithmusbegriff:
 - Programme als Folgen sequentiell auszuführender Anweisungen
- Auf von-Neumann-Rechner ausgerichtet:
 - Anweisungen verändern Daten im Speicher
- Basiskonzepte:
 - Variable: zur Datenspeicherung
 - Anweisung: zur Bearbeitung von Variablen-Werten
- Strukturierungsmethode:
 - Teilalgorithmus, d. h. Prozedur, Methode, Funktion

Programmiermodell: objektorientiert

- Orientiert auf Beschreibung von kooperierenden Objekten:

Objekt =

Semantische Einheit aus

- Daten (Variablen)
- Operationen / Algorithmen zur Bearbeitung der Daten

Klasse = Beschreibungsform für ähnliche Objekte
 Programm = Menge von Klassen

Programmiermodell: logisch

- Programm = auf die logische Beschreibung eines Problems orientiert
- Beschreibung von Relationen zwischen 'Daten' / d.h. zwischen 'Objekten der Realität'
- mittels der Prädikatenlogik

Beispiel:

```

elternteil (lisa, maria).
elternteil (hans, maria).

schwester (X, Y) :-
    elternteil (Z, X),
    elternteil (Z, Y),
    weiblich (X).

weiblich (maria).
weiblich (lisa).

maennlich (hans).
    
```

Problem beschrieben;
nicht: Algorithmus

Programmiermodell: funktional

- Programm = Ein funktionales Programm beschreibt eine Komposition von (mathematischen) Funktionen.
- Mathematische Funktion
 $f: D \rightarrow W$ (Definitionsbereich, Wertevorrat)

- Keine Zuweisung von Werten an Variablen;
- Keine Beschreibung von Algorithmen; z.B. kein Zyklus
- Kein Von-Neumann-Rechner als Hardware-Modell

Beispiel:

```

Programm map, das Funktion f auf alle Elemente einer Liste
anwendet, z.B.
map (add 1) [1,2,3] => [2,3,4]
map (mult 2) [1,2,3] => [2,4,6]
    
```

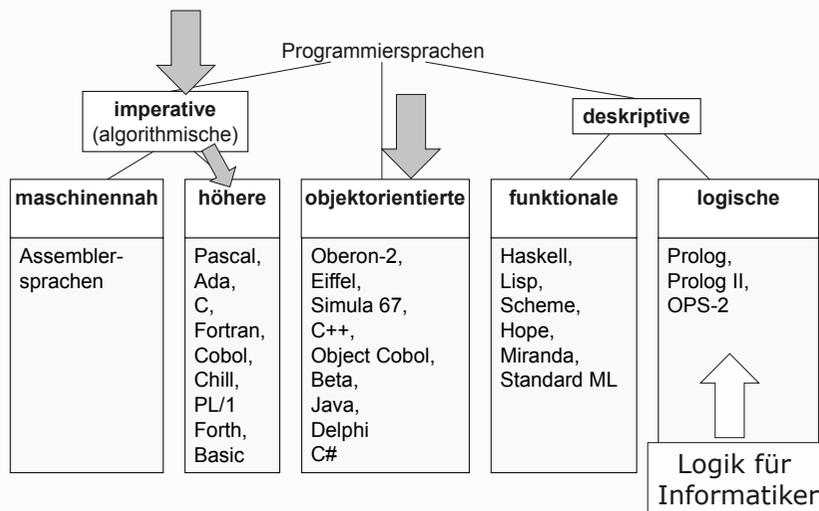
Realisiert durch folgendes funktionales Programm **map**:

```

map f [] = [] (1. Fall: leere Liste [])
map f (x:xs) = f x : map f xs (2. Fall: nicht-leere Liste)
    
```

(Funktion f auf nicht-leere Liste x:xs angewandt, wobei x erstes Element, xs die Restliste ist; „:“ ist Verkettung)

Gegenstand der Vorlesung GdP



Resümee

Es kommt nicht so sehr darauf an, möglichst viele Programmiersprachen zu beherrschen, sondern vielmehr:

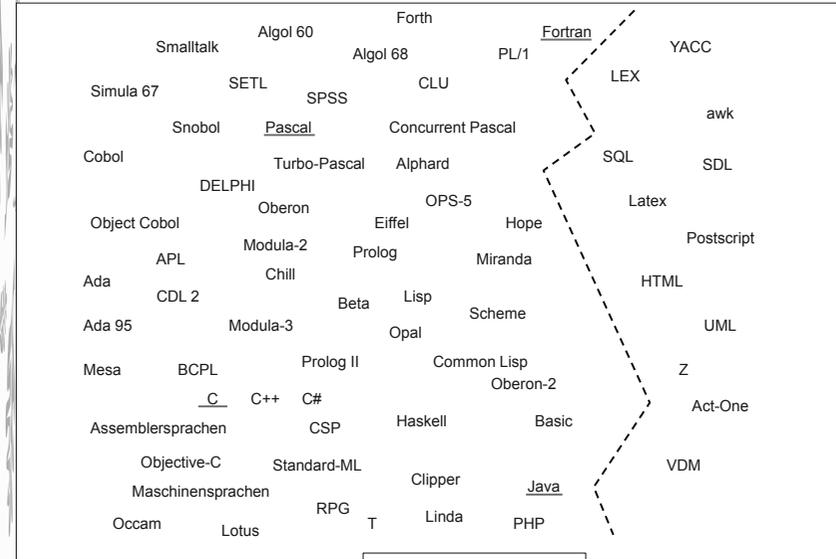
- Konzepte (Anweisungen, Typen ...)
- Programmiertechniken (Sortieren, Suchen ...)
- Komponentenbildung (Modularisierung: Zerlegung von großen Programmen)
- Paradigmen (objektorientiert, imperativ, logisch, funktional)
- Methoden zur Entwicklung komplexer Software (Software Engineering)
- theoretische Grundlagen der Programmierung
- Teamarbeit
- Kenntnisse der Fachdisziplin

➔ Man muss in der Lage sein, sich zielgerichtet in eine neue Sprache einzuarbeiten!

➔ Man sollte Vertreter wichtiger Sprachklassen aktiv beherrschen:
 Java (oder C++, Eiffel, Smalltalk, Delphi), C (oder Pascal), Prolog, Lisp ...

Programmiersprachen: Warum so viele und immer wieder neue?

Warum denn so viele und immer wieder neue Programmiersprachen?



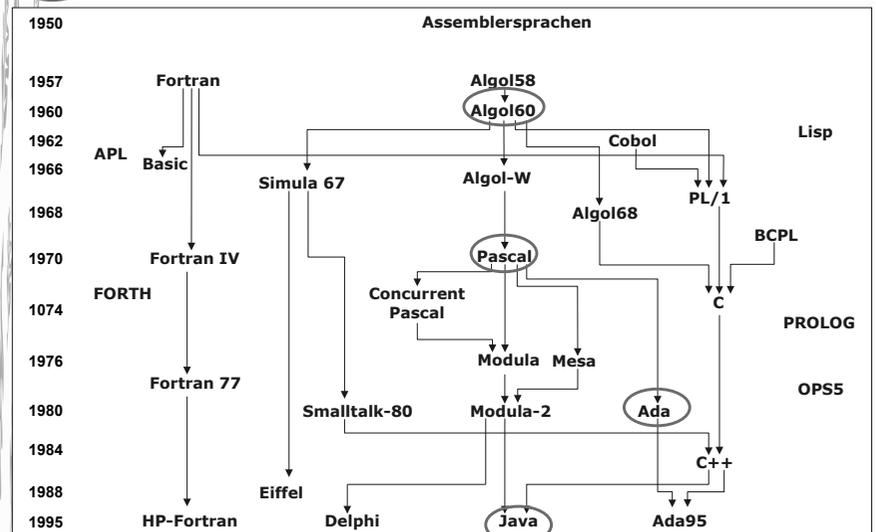
PHP (1997), C# (2000)

Kooperationsprojekte der Gruppe Softwaretechnik (Bothe)

- ▶ Daimler (Stuttgart): Java (Diplomarbeit)
- ▶ Leibniz-Institut:
Fortran 77, 90, 95 (zwei Diplomarbeiten)
- ▶ Institut für Physik der HU: C++
- ▶ Hahn-Meitner-Institut: Turbo-Pascal, LabVIEW
(visuelle Programmierung)
- ▶ Siemens: C#
- ▶ Institut für Psychologie der HU: Smalltalk

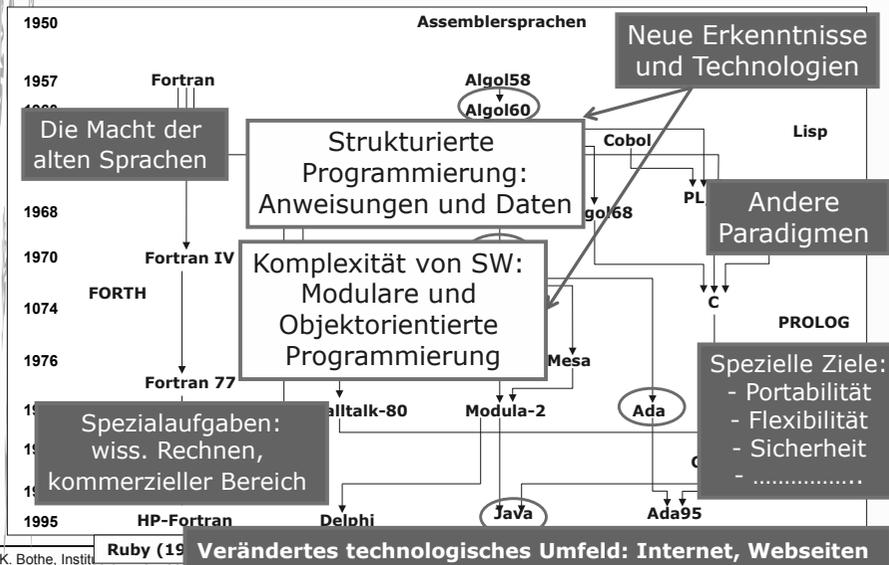
Warum denn so viele und immer wieder neue Programmiersprachen?

○ Kandidaten für DIE endgültige Sprache – in der Vergangenheit



Ruby (1995), Python (1995), PHP (1997), C# (2000), Swift (Apple, 2014), Cuda (2015)

Alte Sprachen und immer wieder neue Sprachen: neue Programmiertechnologien und -aufgaben



Die Macht der alten Sprachen

Fakten zu Cobol

- 1960 entwickelt
- Anwendung: kommerzieller Bereich (Banken, Verwaltung)
- weltweit mehr als 200 Milliarden LOC Cobol-Code im Einsatz
- pro Jahr: 5 Milliarden LOC neu geschriebener Cobol-Code
- 86 % aller Geschäftsdaten durch Cobol-Systeme verarbeitet

Gründe:

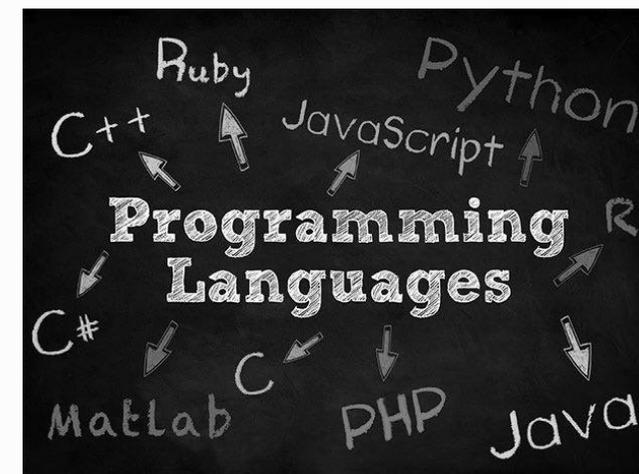
- Reife von Cobol durch ihre Weiterentwicklung,
- Spezialalgorithmen für kaufmännische Berechnungen,
- Routine

(Quelle: Computer Zeitung, 3. März 2003)

Klassifikationen

... nach Verbreitungsgrad

The 2015 Top Ten Programming Languages *)



*) Quelle: IEEE Spectrum, Juli 2015

http://spectrum.ieee.org/computing/software/the-2015-top-ten-programming-languages/?utm_source=techalet&utm_medium=email&utm_campaign=072315

The 2015 Top Ten Programming Languages



“What are the most popular programming languages? The only honest answer: It depends. Are you trying to land a job at a hot mobile app startup, model electricity flows across a continent, or create an electronic art project? Languages are tools, and what’s a “must have” in one domain can be a “whatever” in another.”

- “Java and C are in first place,
- data-analysis language R is rising in popularity,
- newcomers like Apple's Swift and Nvidia's Cuda are making waves.”

Quelle: IEEE Spectrum, Juli 2015

The 2015 Top Ten Programming Languages

Language Rank	Types	Spectrum Ranking ⁽²⁰¹⁵⁾	Spectrum Ranking ⁽²⁰¹⁴⁾
1. Java	Web, Mobile, Enterprise	100.0	100.0
2. C	Mobile, Embedded	99.9	99.3
3. C++	Mobile, Embedded	99.4	95.5
4. Python	Web, Enterprise	96.5	93.5
5. C#	Web, Mobile, Enterprise	91.3	92.4
6. R	Enterprise	84.8	84.8
7. PHP	Web	84.5	84.5
8. JavaScript	Web, Mobile	83.0	78.9
9. Ruby	Web, Enterprise	76.2	74.3
10. Matlab	Enterprise	72.4	72.8

Web
 Mobile
 Enterprise
 Embedded

Quelle: IEEE Spectrum, Juli 2015