



Übung - Modellierung & Programmierung II

Mathias Goldau, Stefan Koch, Wieland Reich

math@informatik.uni-leipzig.de
stefan.koch@informatik.uni-leipzig.de
reich@informatik.uni-leipzig.de



- Anmeldung:
 1. Lernserver: <http://informatik.uni-leipzig.de/ifi/studium/lernserver.html>
 2. Moodle: <https://moodle2.uni-leipzig.de/>
Einschreibeschlüssel: mup2_0XX, $XX \in \{01, 02, \dots, 10\}$
- Problem? Neuigkeiten auf Lehrstuhlseite!
Immernoch? Übungsleiter kontaktieren!
- Übungsserien via Moodle:
 - 5 Serien je 4 Aufgaben + 1 Zusatzaufgabe
 - Abgabe per Moodle
 - Alle Lösungen in ein Archiv zusammenfassen (bspw. ZIP)
 - Alle Gruppen haben einen gemeinsamen Ausgabe- und Abgabetermin, jeweils zu Beginn und Ende des Übungsblocks
 - Besprechung im nächsten Seminar soweit als möglich.
 - Problem mit Kontrolle? Kontrolleur kontaktieren!
Immernoch? Übungsleiter kontaktieren!



Organisatorisches

- Zulassungsvoraussetzung für Klausur: mindestens 30 Punkte
- Konsultationen zu Sprechstunden
- Neugigkeiten auf Lehrstuhlseite: [Lehre > MuP2](#)



Outline

1 Einleitung



- Dienen zur Kommunikation von Menschen mit einem Computer
- Synthetisch und exakt
- Es gibt mehr als 8500 Programmiersprachen ¹
- Historischer Abriss:
http://en.wikipedia.org/wiki/History_of_programming_languages
- Gibt es die “beste” Programmiersprache?
- Eine Sprache ist Turing-Vollständig *gdw* Sie alle Funktionen berechnen kann welche auch eine Turing-Maschine berechnen kann.²
- Die meisten Programmiersprachen sind Turing-Vollständig

¹Thomas J. (Tim) Bergin. “A history of the history of programming languages”. In: *Com. ACM* 50.5 (2007), S. 69–74

²Vgl. Berechenbarkeitstheorie.



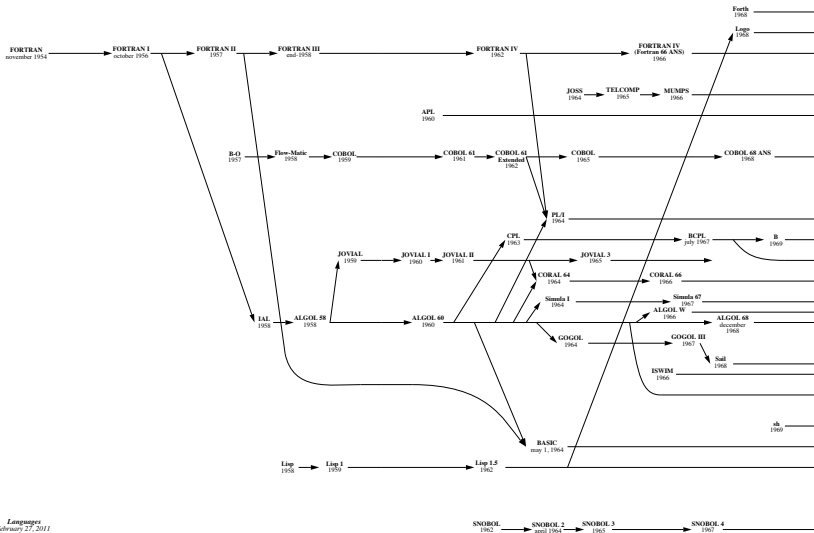
<http://www.levenez.com/lang/>

1954

1957

1960

1965

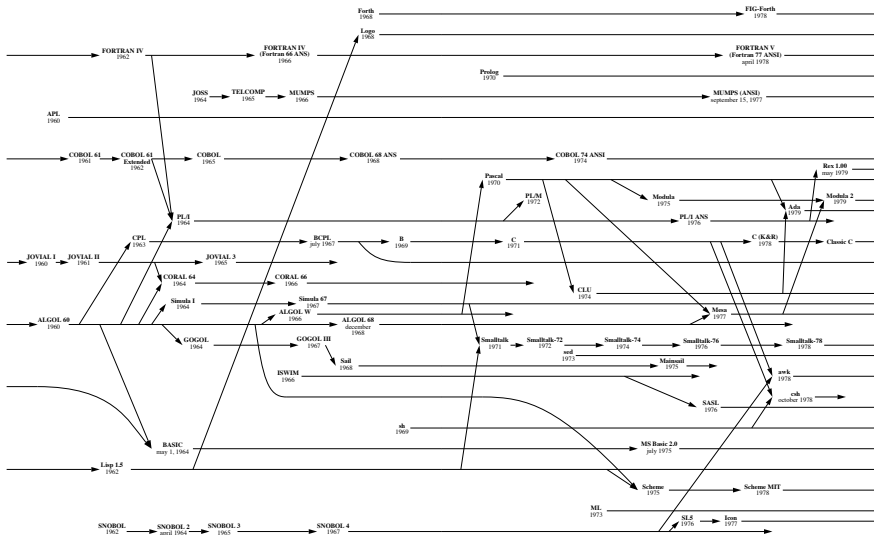


1960

1965

1970

1975

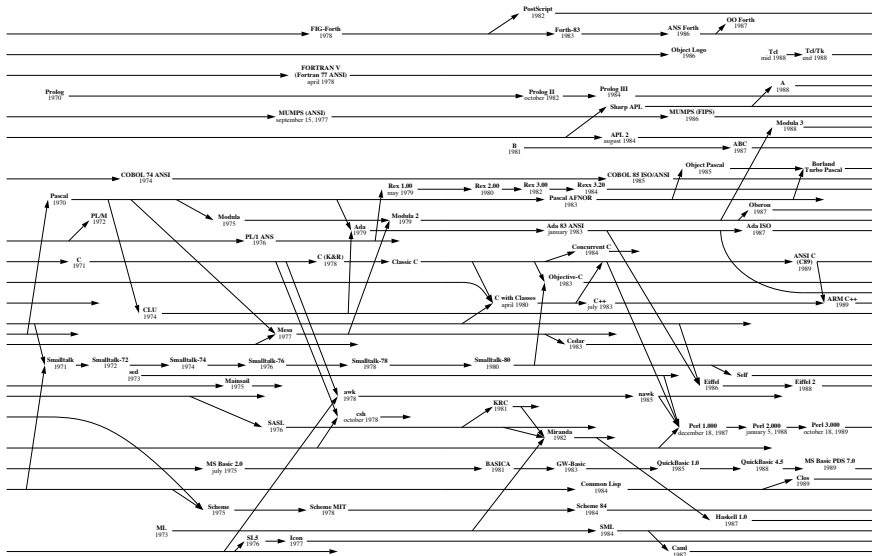


1970

1975

1980

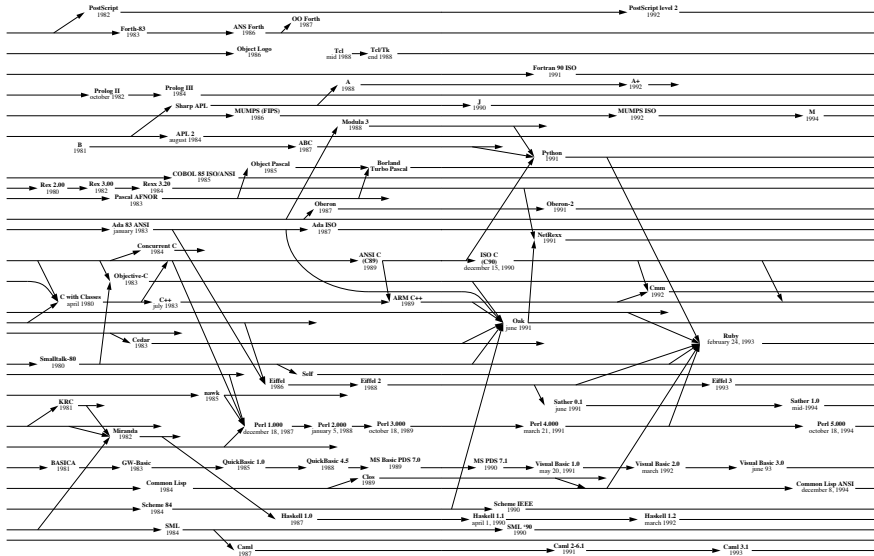
1985



1980

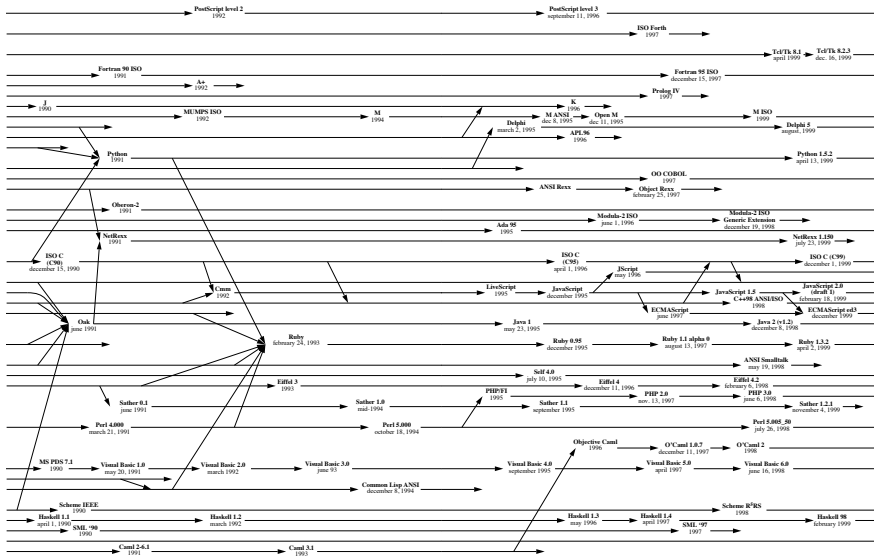
1985

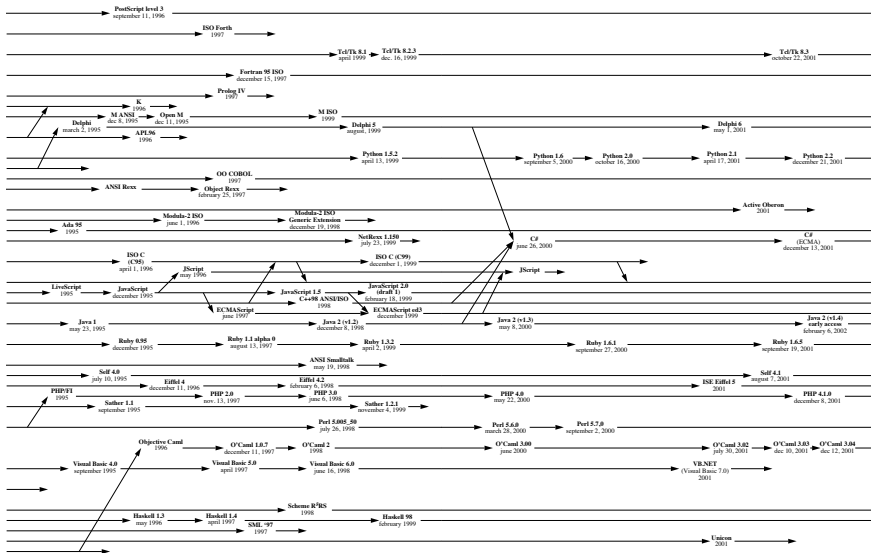
1990

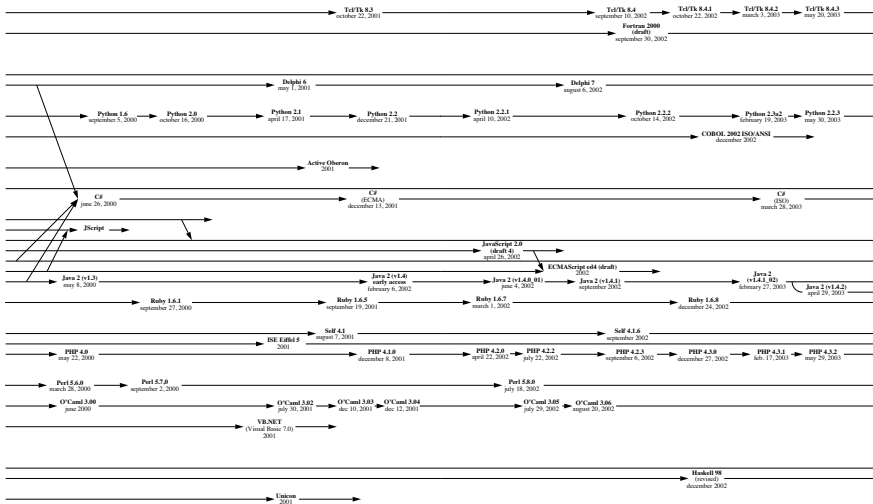


1990

1995



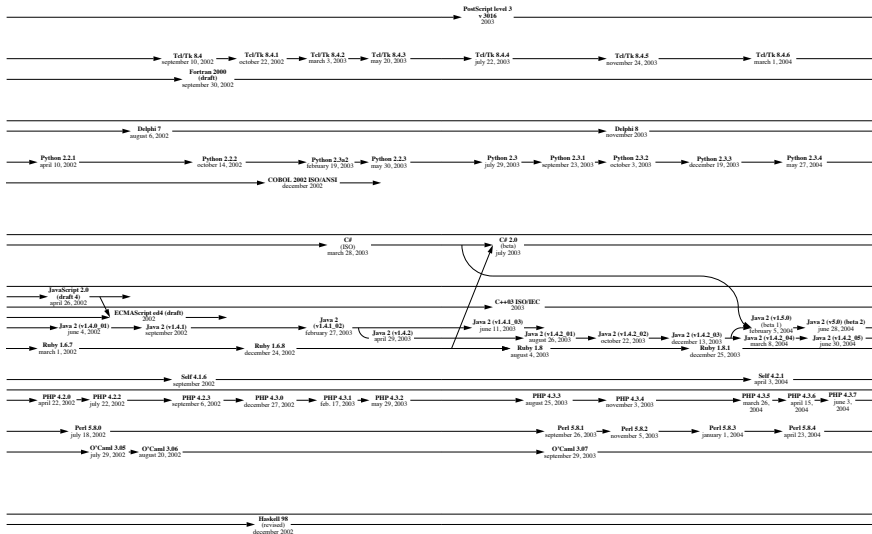




2002

2003

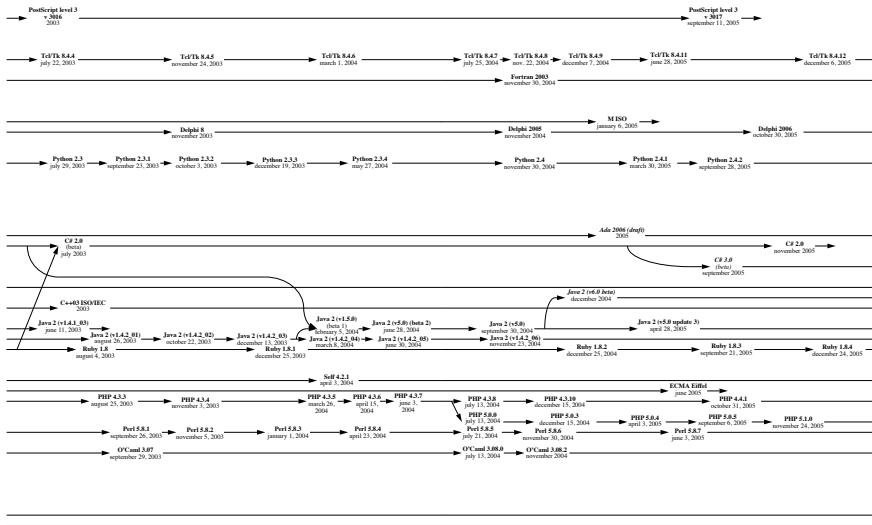
2004



2003

2004

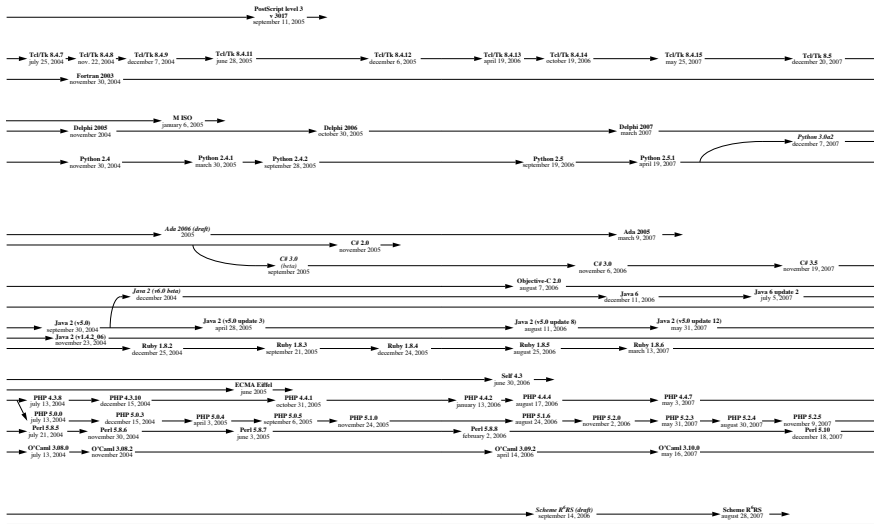
2005



2005

2006

2007

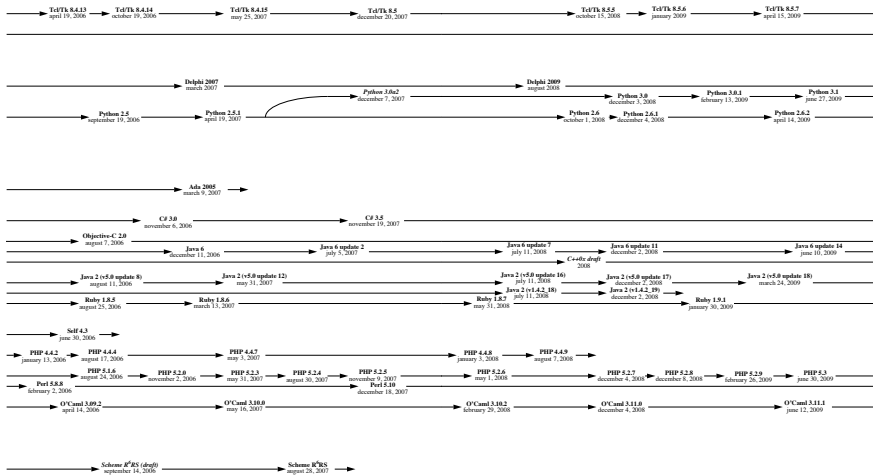


2006

2007

2008

2009

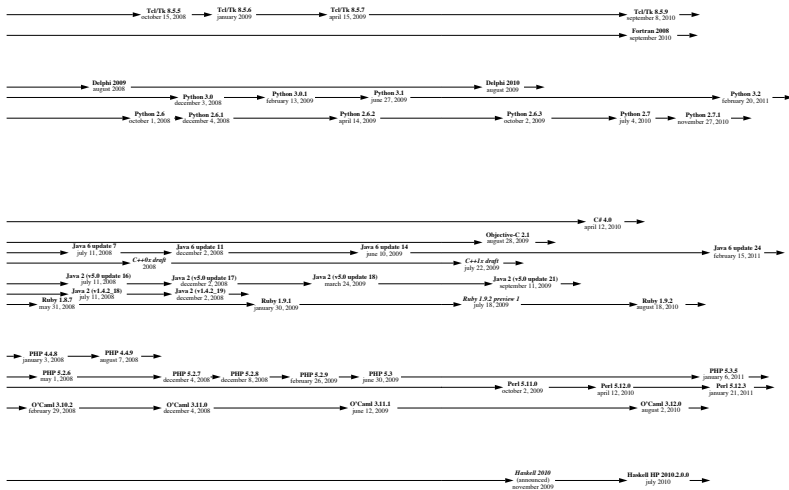


2008

2009

2010

2011





- Imperative: Beschreibt das *WIE*
 - Strukturiert (`if`, `else`,... aber kein `GOTO`)
 - Prozedural (Es gibt Prozeduren)
 - Modular (Prozeduren in Module zusammengefasst)
 - Abstrakte Datentypen (bspw. Objektorientiert)
- Deklarative: Beschreibt das *WAS*
 - Funktional
 - Logikbasiert
 - Bedingungsbasiert (Vor- und Nachbedingungen)



Einstieg in eine (neue) Programmiersprache

- Oftmals ein minimales Programm bspw. Hello World!³
- Daher zum Einstieg am Besten *keine* Integrierte Entwicklungsumgebung (IDE), sondern ein Texteditor!
- Hello World! nicht immer direkt möglich, bspw. OpenGL
- Unterschiede zwischen Sprachen: Syntax, Anweisungen, Variablen, Operatoren, Datentypen, Bibliotheken

³http://en.wikipedia.org/wiki/List_of_Hello_world_program_examples



Mehrsprachige Programme (polyglott)

```
1 /*The following is in C, Pascal, FORTRAN, and PHP.*/
2
3 const a = '\'; void b()/*'; var b:string;{
4 c */ { /*
5 c document.fgColor='#ffffff'";
6 cos(1);print "Merry Christmas" ?>
7 17 format('Merry Christmas')
8 write(6, 15)
9 stop
10 end
11 c */
12 char *a = "}begin b\coloneqq'{"; } int main () { /*'; writeln{*/
13 char cbuf[64]; sprintf(cbuf, "{'Merry Christmas') end. {"");
14 cbuf[29] = '\0'; printf(cbuf+3); return 0; }
```

<http://www.gnu.org/fun/jokes/merry-xmas.html>



Mehrsprachige Programme (polyglott)

```
1 /*The following is in C, Pascal, FORTRAN, and PHP.*/
2
3 const a = '\'; void b()/*'; var b:string;{
4 c */ { /*
5 c document.fgColor='#ffffff';
6 cos(1);print "Merry Christmas" ?>
7 17 format('Merry Christmas')
8 write(6, 15)
9 stop
10 end
11 c */
12 char *a = "}begin b\coloneqq'{"; } int main () { /*'; writeln{*/
13 char cbuf[64]; sprintf(cbuf, "{'Merry Christmas') end. {"");
14 cbuf[29] = '\0'; printf(cbuf+3); return 0; }
```

<http://www.gnu.org/fun/jokes/merry-xmas.html>



Mehrsprachige Programme (polyglott)

```
1 /*The following is in C, Pascal, FORTRAN, and PHP.*/
2
3 const a = '\'; void b()/*'; var b:string;{
4 c */ { /*
5 c document.fgColor='#ffffff';
6 cos(1);print "Merry Christmas" ?>
7 17 format('Merry Christmas')
8 write(6, 15)
9 stop
10 end
11 c */
12 char *a = "}begin b\coloneqq'{"; } int main () { /*'; writeln{*/
13 char cbuf[64]; sprintf(cbuf, "{'Merry Christmas') end. {"");
14 cbuf[29] = '\0'; printf(cbuf+3); return 0; }
```

<http://www.gnu.org/fun/jokes/merry-xmas.html>



Mehrsprachige Programme (polyglott)

```
1 /*The following is in C, Pascal, FORTRAN, and PHP.*/
2
3 const a = '\'; void b()/*'; var b:string;{
4 c */ { /*
5 c document.fgColor='#ffffff';
6 cos(1);print "Merry Christmas" ?>
7 17 format('Merry Christmas')
8 write(6, 15)
9 stop
10 end
11 c */
12 char *a = "}begin b\coloneqq'{"; } int main () { /*'; writeln{*/
13 char cbuf[64]; sprintf(cbuf, "){('Merry Christmas') end. {"");
14 cbuf[29] = '\0'; printf(cbuf+3); return 0; }
```

<http://www.gnu.org/fun/jokes/merry-xmas.html>



Mehrsprachige Programme (polyglott)

```
1 /*The following is in C, Pascal, FORTRAN, and PHP.*/
2
3 const a = '\'; void b()/*'; var b:string;{
4 c */ { /*
5 c document.fgColor='#ffffff'";
6 cos(1);print "Merry Christmas" ?>
7 17 format('Merry Christmas')
8 write(6, 15)
9 stop
10 end
11 c */
12 char *a = "}begin b\coloneqq'{"; } int main () { /*'; writeln{*/
13 char cbuf[64]; sprintf(cbuf, "){('Merry Christmas') end. {"");
14 cbuf[29] = '\0'; printf(cbuf+3); return 0; }
```

<http://www.gnu.org/fun/jokes/merry-xmas.html>