Programmiersprachen
Vorlesung 1 vom 21.10.21
Einführung

Christoph Lüth

Universität Bremen

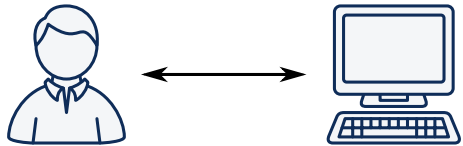
Wintersemester 2021/22

Einführung

Worum geht es?

- Es gibt über 700 Programmiersprachen (sagt Google)
- Wie kriegen wir da Ordnung rein?
- Zugrundeliegende Prinzipien
  - Was haben alle Programmiersprachen gemein?
  - Wo gibt es Unterschiede?
  - **Taxonomie** der Programmiersprachen
- Neue Programmiersprachen lernen (neue, alte, merkwürdige)

Warum Programmiersprachen?



- Wollen Programme in **verständlicher** Notation aufschreiben
- Maschine soll sich dem Menschen anpassen (nicht umgekehrt)
- Programme müssen **maschinenlesbar** und **auführbar** bleiben
- **Modellbildung** und **Abstraktion**

Konzept der Veranstaltung

- Vorlesung:
  - Montag um 10:00, online
    - Zoom: `https://uni-bremen.zoom.us/j/93767566115?pwd=QThKSlR0ckVLdUpGc3BqMm5wNTN6Zz09`
  - Dazu Übungsblatt
- Übungen:
  - Werden bis Donnerstag bearbeitet
  - Lösungen werden am Donnerstag 8-10 in der Übung vorgestellt
  - Dazu "Musterlösung" vom Veranstalter
  - Werden **nicht korrigiert**
- Referate:
  - Ab Woche 10/11
  - Studierende stellen je **eine** neue Sprache vor

Scheinbedingungen

- Referat über eine neue Sprache
- Mündliche Prüfung am Ende

Grundlagen

Was ist eine Programmiersprache?

1. Definierte, maschinenlesbare **Syntax**
2. Mathematisch, informell oder pragmatisch definierte **Semantik**
3. Die Sprache muss **ausführbar** und **Turing-mächtig** sein

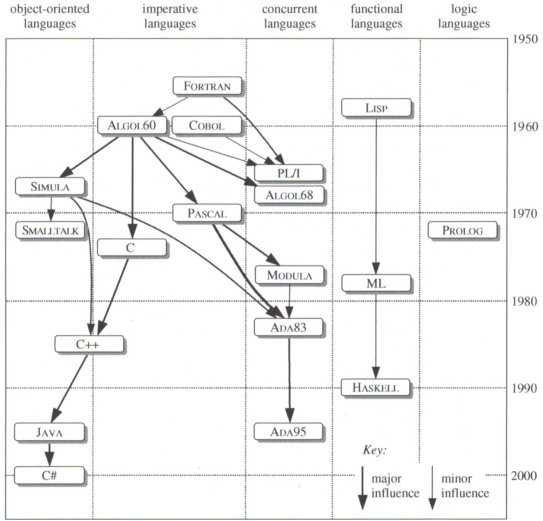
Arten von Programmiersprachen

- Programmiersprachen sind immer **Abstraktionen** über einem Berechnungsmodell.
- **Imperativ**: Zustandsübergänge auf einem Speicher (Turing-Maschine)
  - Abstraktion durch Datentypen
  - Abstraktion durch Verkapselung
- **Funktional**: Rekursive Funktionen (Auswertung von Ausdrücken)
- Sonstige: logische, domänenspezifisch

Scope dieser Veranstaltung — was machen wir nicht?

- Aspekte der Ausführung: wie **implementieren** wir eine Programmiersprache
  - Compilerbau, Übersetzer, abstrakte Maschinen, ...
- Aspekte der Syntax
  - Grammatiken und formale Sprachen, Parsergeneratoren, Lexer, ...
- Formale Semantik
  - Mathematische Beschreibung der Semantik, semantische Rahmenwerke, Typentheorie

Historisches: Stammbaum einiger Programmiersprachen



Welche Sprachen betrachten wir?

- Laufende Beispiele:
  - C
  - Java
  - Python
  - Haskell
- Weitere in den Referaten

Liste weiterer Sprachen

- Systemnah: Rust
- Logische Programmierung: Prolog, Oz
- Dynamisch: JavaScript
- Nebenläufig/Reaktiv: Erlang, Golang
- Abhängige Typen: Idris, Agda, Liquid X (Dependent types)
- Prozedural: Julia, Kotlin, Swift
- Skriptsprachen: Lua, Tcl, sh/bash
- Funktional[1]: SML, OCAML, Elm, Clojure, LISP, Scala
- Stack-basiert: Forth
- Historisch: COBOL, Algol-68, APL, Ada, Smalltalk
- Datenflusssprachen: Id, Lucid, Lustre
- DSLs: R, SQL, Postscript, TeX, Verilog/VHDL, SystemC, SpinalHDL

[1] Optional

Struktur der Veranstaltung

- Einführung
- Werte und Typen
- Anweisungen, Variablen und Zustand
- Kontrollabstraktion
- Datenabstraktion
- Fortgeschrittene Typsysteme
- Nebenläufigkeit
- Objektorientierung
- Skriptsprachen
- Beispielsprache II
- Ab Woche 11: Studentische Vorträge.

Literatur und Basis

- David A. Watt: **Programming Language Design Concepts**, John Wiley & Sons, 2004.
- Maurizio Gabbrielli, Simone Martini: **Programming Languages: Principles and Paradigms**. Springer, 2010.
- Robert W. Sebesta: **Concepts of Programming Languages**. Pearson Education, 2016.

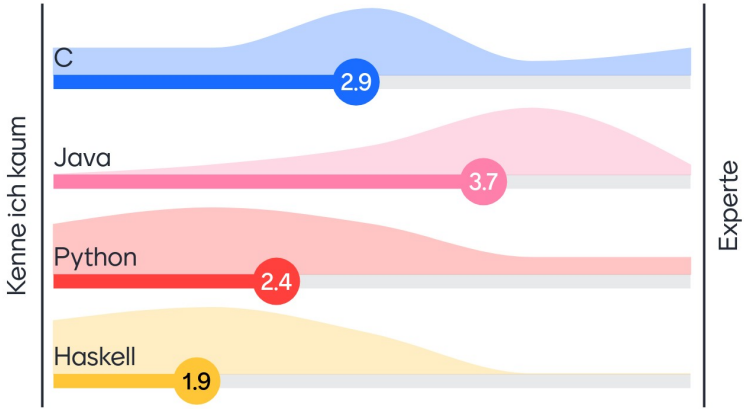
Vorkenntnisse

Online-Umfrage: `https://www.menti.com/veswebppdp`

- Vorkenntnise in folgende Sprachen:
  - C
  - Java
  - Python
  - Haskell
- Welche weiteren Sprachen kennt ihr?

Vorkenntnisse

Mentimeter



12

Welche anderen Sprachen kennt ihr?

Mentimeter

12

Zum Abschluss

Eine einfache Funktion

In den Programmiersprachen C, Haskell, Java, Python:
Schreibe eine Funktion (Methode), welche als Argument eine Zeichenkette bekommt, und zählt, wie oft die Zeichen `x`, `y` und `z` (egal, ob groß oder klein) auftreten.

Zum Abschluss

Eine einfache Funktion

In den Programmiersprachen C, Haskell, Java, Python:
Schreibe eine Funktion (Methode), welche als Argument eine Zeichenkette bekommt, und zählt, wie oft die Zeichen `x`, `y` und `z` (egal, ob groß oder klein) auftreten.

Nächster Termin

- Montag, 25.10.2020 um 10 ct auf Zoom.