

EINFÜHRUNG IN DIE PROGRAMMIERUNG

FUNDAMENTALS

DHBW MANNHEIM

WIRTSCHAFTSINFORMATIK (DATA SCIENCE)

Markus Menth

Martin Gropp

MOTIVATION

Ziel: Computern Instruktionen geben

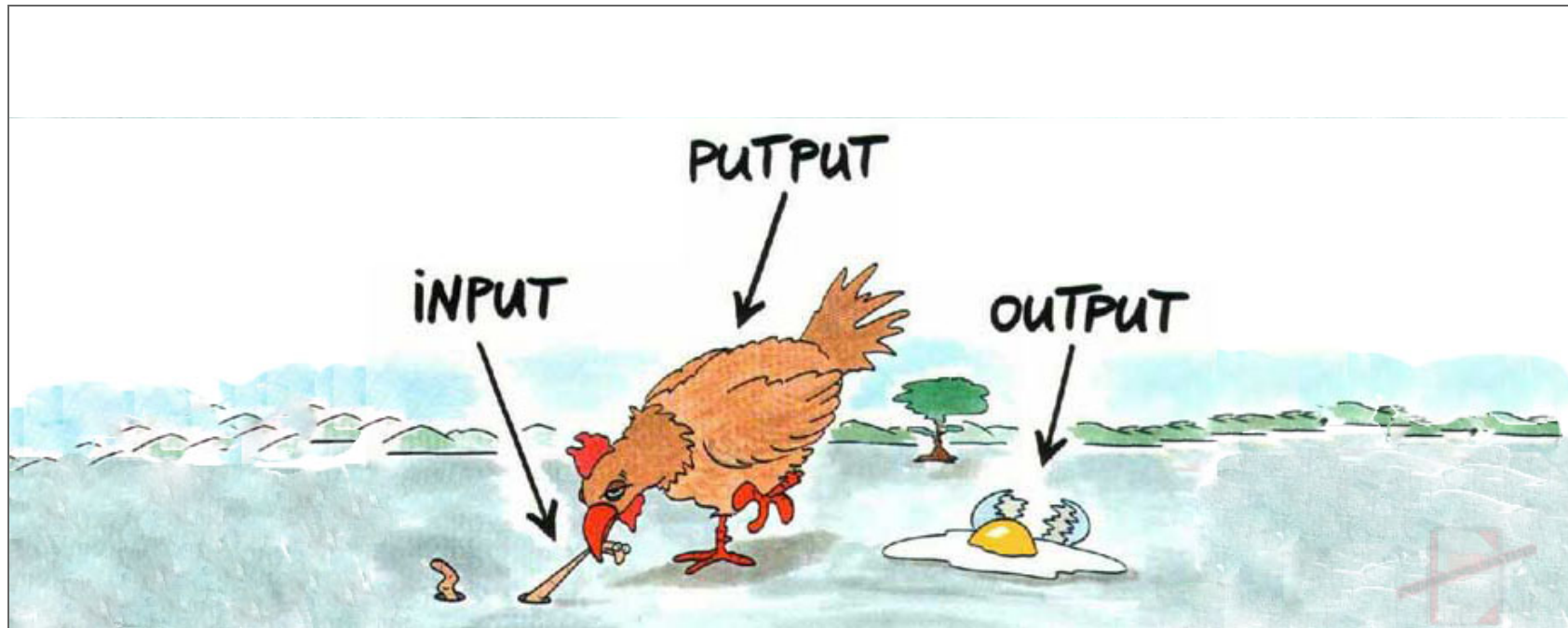


Image source: programmingnotes.org

INSTRUKTIONEN: MASCHINENSPRACHE

00000000:	EA	05	00	C0	07	B4	00	CD	16	2C	30	00	C0	04	30	88
00000010:	C1	B4	0E	BB	07	00	B0	0D	CD	10	B0	20	CD	10	CD	10
00000020:	B0	0D	CD	10	88	C8	3C	30	72	DB	3C	3A	72	0C	3C	45
00000030:	73	D3	B0	31	CD	10	88	C8	2C	0A	CD	10	EB	C7	00	00

Sprungbefehl: Lade Programmzähler mit Adresse 07C0:0005

Datenregister mit Konstante laden: Schreibe 0 in das Register ah

Ein-/Ausgabewerk benutzen: Signal „16“ bedeutet bei ah=0: Lese einen Tastendruck ein. Ergebnis (ASCII-Wert) wird in das Register al geschrieben

Subtrahiere Konstante: Subtrahiere 30 vom Wert in al und schreibe Ergebnis nach al

Addiere: Rechne „Wert in al“ + „Wert in al“ und schreibe Ergebnis nach al.

Und so weiter...

MASCHINENSPRACHE \Leftrightarrow MENSCHEN

Maschinensprache schwer lesbar für Menschen

- Daher: höhere Programmiersprache
- Näher am menschlichen Denken

Höhere Programmiersprache

- Wird von Programmen (Compiler/Interpreter) in Maschinencode übersetzt

Es existieren hunderte verschiedener solcher Sprachen

- Welche man wählt, hängt vom Anwendungszweck ab
- Prinzipiell sind alle gleich mächtig (Turing-vollständig)

SPRACHE MIT EIGENER SYNTAX UND SEMANTIK

"Walfische bereisen Indien, um Wolken zu klauen!"

Syntax:

- Grammatikregeln der Sprache

Semantik:

- Bedeutung einzelner "Worte" und "Satzzeichen" der Sprache

Der obige Satz ist syntaktisch korrekt – aber sinnlos

ZIEL DER PROGRAMMIERUNG

Umsetzung eines gegebenen oder selbstentwickelten Algorithmus in ein lauffähiges Computerprogramm

	Problem oder Realität	←	Welches Problem stellt sich in der Realität?
Analyse und Modellierung	↓		
	Algorithmische Beschreibung	←	Wie kann ich das Problem lösen?
Programmierung und Codierung	↓		
	Programm	←	Welche Programmiersprache ist geeignet?
Übersetzung und	↓		

Compilierung		
	Ausführbares Programm	← Wie sieht der ausführbare Code aus?
Ausführung und Interpretierung	↓	
	Problemlösung?	← Have we built the program right? Have we built the right program?
Testen		

BEISPIELE FÜR ALGORITHMEN

- Bedienungsanleitungen
- Bauanleitungen
- Kochrezepte
- Mathematische Problemstellungen
- Such- und Sortieralgorithmen

BEISPIEL: KOCHREZEPT

Tiramisu mit Spekulatius und Himbeeren (37 Portionen)

Menge	Zutat
3053 ml	Schlagsahne (Obers)
2313 g	Mascarpone
2313 g	Quark (Topfen)
833 g	Zucker, braun
9 1/4 TL	Vanillezucker
3700 g	Spekulatius
4625 g	Himbeeren, TK, auftauen und abtropfen

Anleitung

Schlagobers fest schlagen. Mascarpone, Topfen, Zucker und Vanillezucker einrühren. In einer rechteckigen Form den Boden mit Creme bestreichen, eine Schicht Spekulatius auflegen und in die Creme drücken, darüber eine

Anleitung

Schicht Beeren verteilen, Creme, Spekulatius (wieder etwas in die Creme drücken) ... die letzte Schicht soll Creme sein, diese mit Kakaopulver bestreuen. Kalt stellen für ein paar Stunden. Die Spekulatius werden weich.

Welche Probleme ergeben sich bei der Umsetzung des Rezeptes?

PROBLEME

Beschreibung lässt Raum für individuelle Entscheidungen und Interpretationen

Ungenaue Aussagen

- fest schlagen
- etwas in die Creme drücken
- mit Kakaopulver bestreuen
- Kakaopulver fehlt in der Liste der Zutaten

Verbesserungsmöglichkeiten

- komplette Eliminierung von individuellen Interpretationsspielräumen
- vollständige Beschreibung der Arbeitsschritte

Daher: exakt definierte Sprache und Operatoren notwendig

DEFINITION DES ALGORITHMUSBEGRIFFS

Duden

- Algorithmus (lat. algorismus = Art der indischen Rechenkunst, in Anlehnung an griech. arithmós = Zahl entsteht aus dem Namen des pers.- arab. Mathematikers Al-Hwarizmi, gest. nach 846)
- Verfahren zur schrittweisen Umformung von Zeichenreihen; Rechengvorgang nach einem bestimmten [sich wiederholenden] Schema.

Informatik

- Ein Algorithmus ist eine präzise (d.h. in einer festgelegten Sprache abgefasste) endliche Beschreibung eines allgemeinen Verfahrens unter Verwendung ausführbarer elementarer (Verarbeitungs-)Schritte

BESTANDTEILE VON ALGORITHMEN

Elementare Operationen (→ sog. Operatoren)

- Addition, Subtraktion, etc.: Berechne 1 plus 2

Sequenz

- Hintereinander-Ausführung von Anweisungen: Tu das, dann das,
...

Bedingte Ausführung

- Beispiele: Wenn die Ampel grün ist fahre weiter oder Wenn die Ampel rot ist, halte an.

BESTANDTEILE VON ALGORITHMEN

Wiederholungs-Anweisung

- Beispiel: Zähle von 1 bis 10

Speicherplatz

- Variablen (und Konstanten)

EIGENSCHAFTEN VON ALGORITHMEN

Terminierung

- Bricht nach endlich vielen Schritten ab

Determinismus

- Deterministischer Ablauf
 - Legt eindeutige Vorgabe der Schrittfolge der auszuführenden Schritte fest
 - Nicht jeder Lauf muss identisch sein
- Determiniertes Ergebnis
 - Wird immer dann geliefert, wenn bei vorgegebener Eingabe ein eindeutiges Ergebnis geliefert wird
 - Auch bei mehrfacher Durchführung mit denselben Eingabeparametern (z.B. kein Zufall enthalten)

KRITERIEN FÜR GUTE PROGRAMME

Did we build the right program?

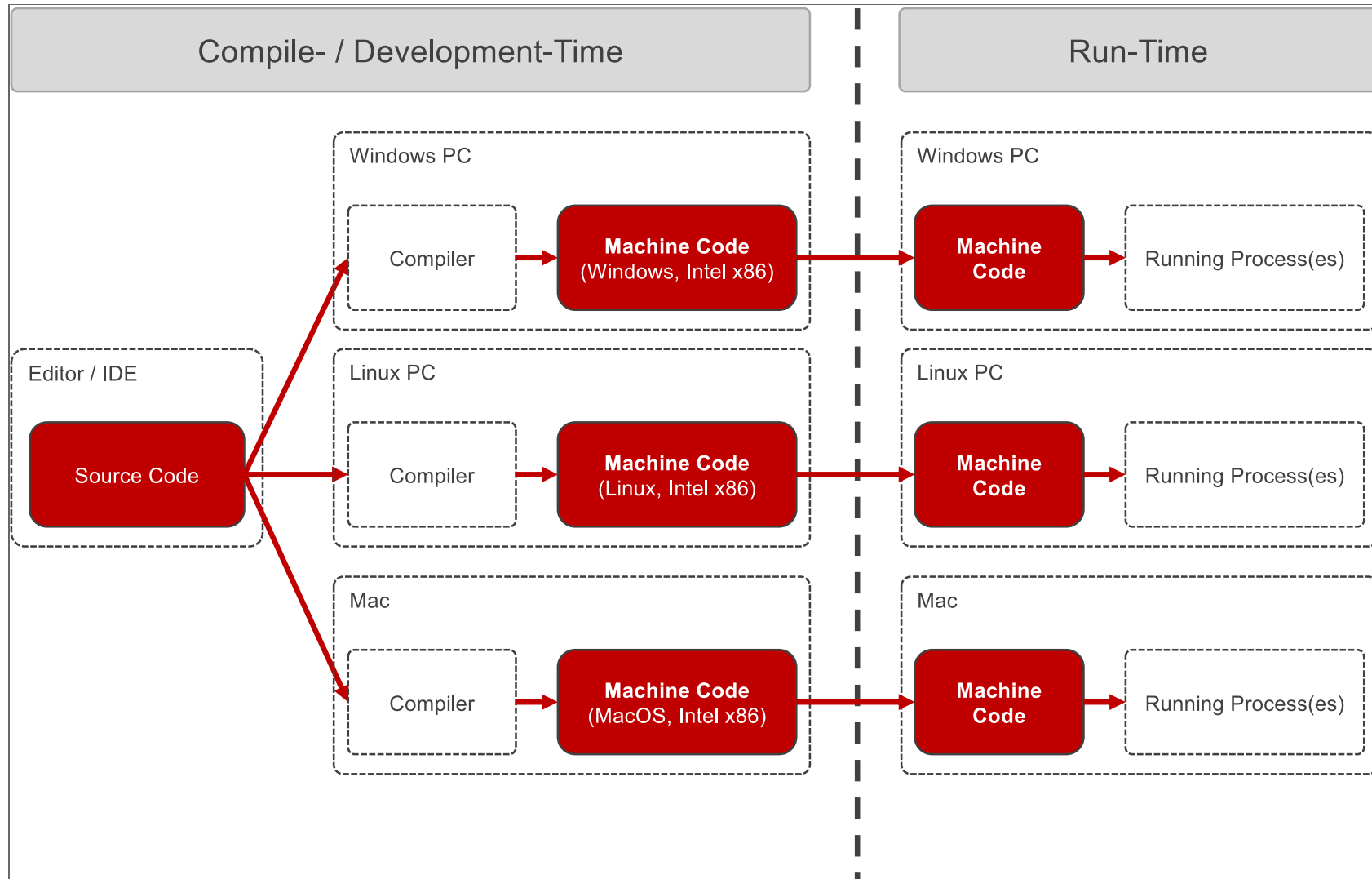
- Erfüllt das Programm die Aufgabenstellung
- Erfüllt es alle funktionalen/nicht-funktionalen Anforderungen

Did we build the program right?

Weitere

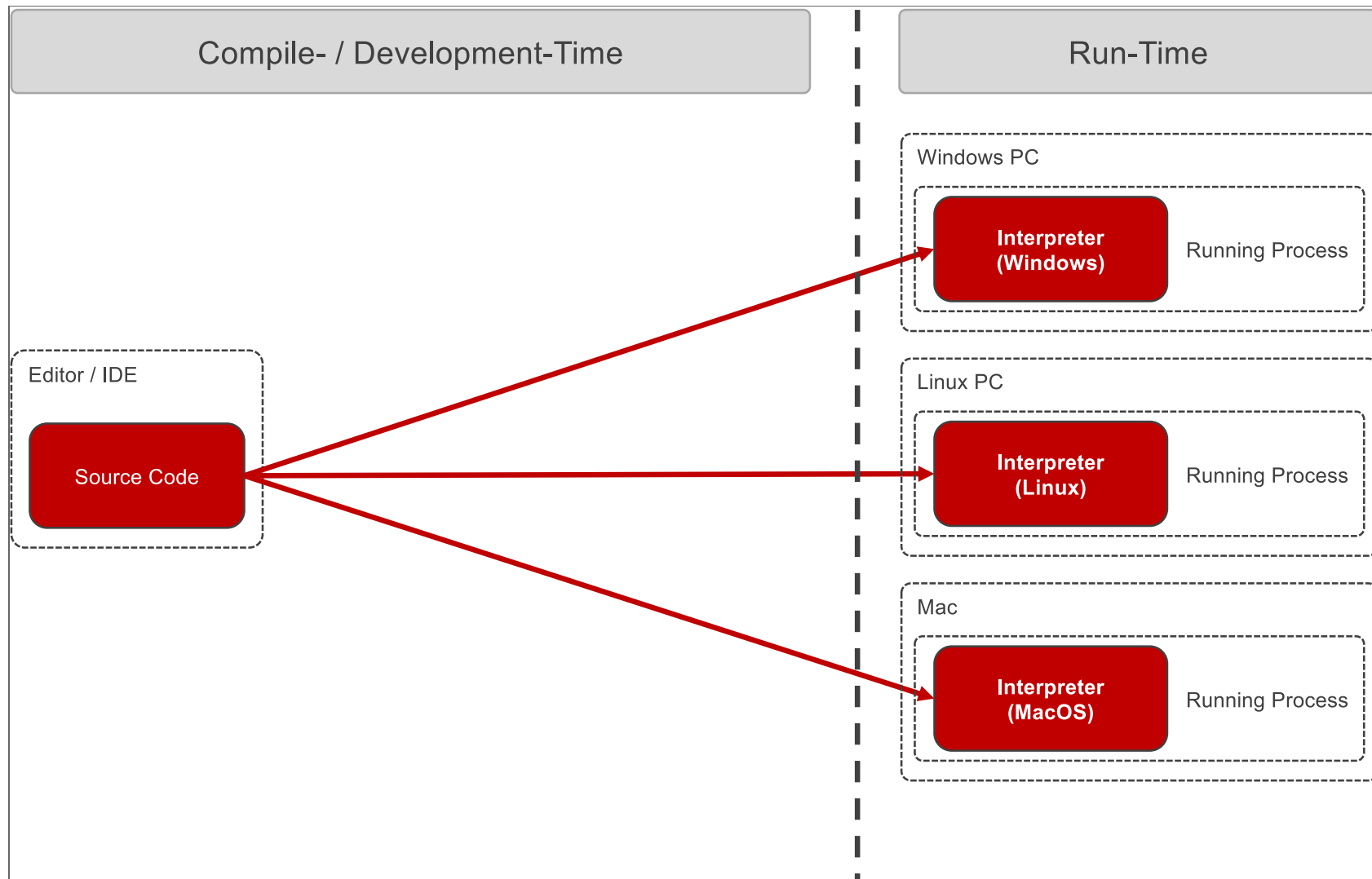
- Effizienz (Speicher/Laufzeit)
- Robustheit
- Sicherheit
- Übertragbarkeit (andere Betriebssysteme, etc.)
- Lesbarkeit, Wartbarkeit, Erweiterbarkeit

COMPILER



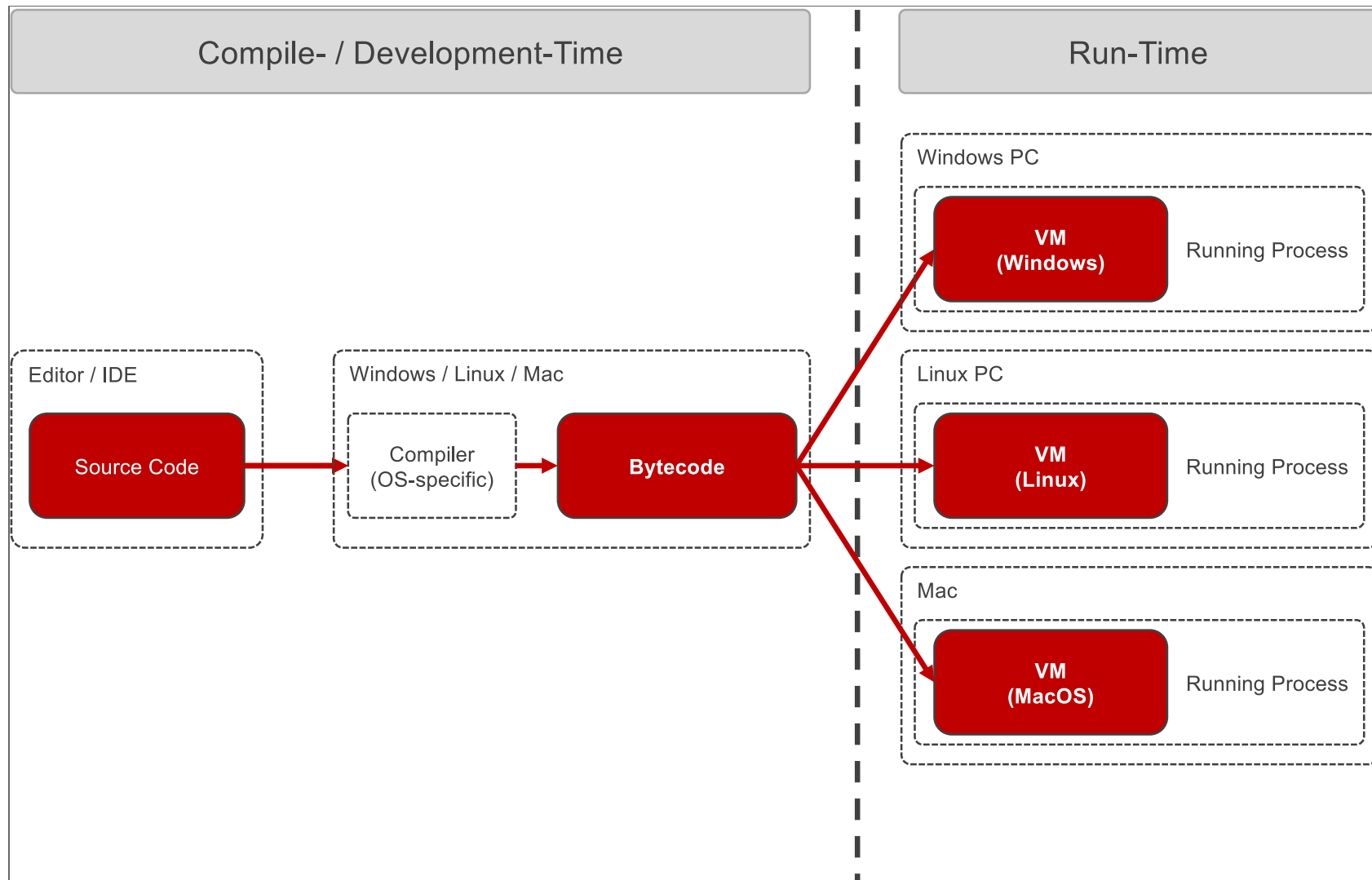


INTERPRETER





BYTECODE





HÖHERE PROGRAMMIERSPRACHEN

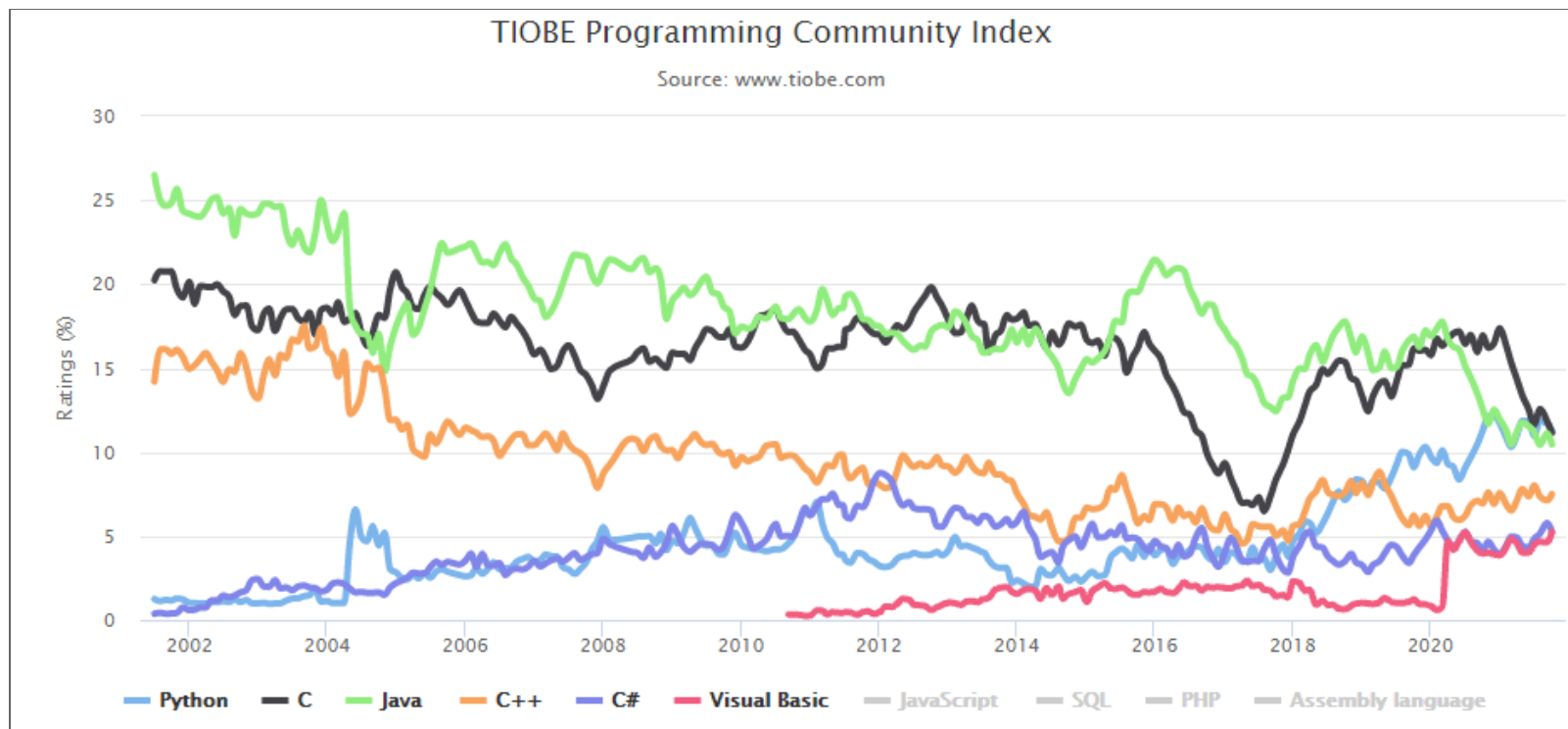
Große Auswahl von Programmiersprachen

- **Java, C, C++, Python, Visual Basic .NET, C#, PHP, JavaScript, SQL, Swift, MATLAB, Go, Assembler, R, Objective-C, Perl, Delphi, Ruby, PL/SQL, Visual Basic, D, SAS, Dart, Scratch, COBOL, Scala, F#, Groovy, Lua, ABAP, Fortran, Lisp, Transact-SQL, Rust, Ada, Logo, LabVIEW, Prolog, Haskell, Scheme, Kotlin, Forth, Julia, Erlang, Ladder Logic, Apex, PL/I, Bash, Clojure, Tcl, ABC, ActionScript, Alice, APL, Awk, BBC BASIC, bc, Bourne shell, C shell, CL (OS/400), Clarion, CoffeeScript, Common Lisp, Crystal, cT, Euphoria, Hack, Icon, Inform, Io, J, Korn shell, LiveCode, ML, Modula-2, Monkey, MOO, MQL4, MS-DOS batch, NATURAL, OCaml, OpenCL, OpenEdge ABL, Oz, PILOT, PostScript, b, Q, Racket, Ring, RPG, S, Snap!, SPARK, SPSS, Tex, TypeScript, Vala/Genie, Verilog, VHDL, ...**

BESTE HÖHERE PROGRAMMIERSPRACHE?

Popularität ändert sich über die Zeit

- Vgl. TIOBE Index



HÖHERE PROGRAMMIERSPRACHE

Diese Veranstaltung: Python

- Erscheinungsjahr: 1991
- Üblicherweise interpretiert (mit Bytecode-Compilation)
- Weicht in Details von "klassischen" Programmiersprachen ab
- Beispiel: Strukturierung durch Einrückung statt Klammerung

Eigenschaften von Python

- Unterstützt mehrere Programmierparadigmen (objektorientiert, aspektorientiert und funktional)
- Dynamisch typisiert
- Umfangreiche Standardbibliothek
- De-facto Standard im Data Science-Umfeld