

Programmiersprachen

Jannik Esser

10. September 2022

Inhalt

1. Einleitung

2. Hintergrundwissen

Gebiete der Softwareentwicklung

Unterschiede in Programmiersprachen

3. Programmiersprachen

4. Zusammenfassung

Gebiete der Softwareentwicklung

- ▶ Web development (Webentwicklung)
- ▶ Application development (Anwendungsentwicklung)
- ▶ Data science
- ▶ Artificial Intelligence \supsetneq Machine Learning (künstliche Intelligenz \supsetneq maschinelles Lernen)
- ▶ Game development (Spieleentwicklung)
- ▶ Embedded systems (eingebettete Systeme)
- ▶ Operating systems (Betriebssysteme)

Web- & App development

- ▶ Entwicklung für Enduser
- ▶ Webentwicklung für den Browser bzw. das Internet
- ▶ Applikationsentwicklung für ein oder mehrere Betriebssysteme
- ▶ Die Ziele sind gleich oder ähneln sich
- ▶ Cross-platform development für eine Code base für beide Bereiche

Data science

- ▶ Interdisziplinäres Feld
- ▶ Wissen aus Daten gewinnen

Artificial Intelligence \supsetneq Machine Learning

- ▶ Viele Konzepte und Algorithmen inbegriffen
- ▶ Maschinelles Lernen ist ein Teilbereich
- ▶ Verwendet in: Alexa, Siri, ...
- ▶ Viele Firmen sind interessiert in Algorithmen dieser Art

Game development

- ▶ Viele Sprachen
- ▶ Sehr beliebt
- ▶ Nicht die beste Branche was Bezahlung angeht

Embedded systems

- ▶ Unsichtbare Aufgaben
- ▶ Abdecken von verschiedensten Aufgaben
- ▶ Vernetzen von verschiedenen autonomen Systemen

Operating systems

- ▶ Management von Systemressourcen
- ▶ Schnittstelle zwischen Hardware und Software
- ▶ Beispiele: Windows, Linux, MacOS, Android, iOS, ...

Paradigmen

- ▶ Unterschiedliche Arten Funktionalitäten darzustellen
- ▶ Prozedural, objektorientiert, funktional

Prozedural

- ▶ Älteste Form des strukturierten Programmierens
- ▶ Programm flow von oben nach unten
- ▶ Abstraktionen durch Funktionen
- ▶ Zusammenhalten von Daten durch Strukturen möglich

Objektorientiert

- ▶ Zusammenfassen von Daten als Objekte
- ▶ Objekte haben Eigenschaften und Verhalten
- ▶ Prozedurale Eigenschaften vorhanden durch Programmflow von oben nach unten

Funktional

- ▶ Erinntert an prozedurale Programmierung
- ▶ Zustand von Strukturen wird nicht verändert
- ▶ Funktionen erstellen Kopien von ursprünglichen Strukturen mit Veränderungen
- ▶ $x_1 = x_2 \Rightarrow f(x_1) = y$ und $f(x_2) = y$
- ▶ Beliebtheit steigt

Typsicherheit

- ▶ Fehler Entdeckung bei inkompatiblen Typen
- ▶ Typen sind z.B. `int` und `String`

Typsicherheit

```
int x = 3;  
String str = "Hello There";  
System.out.println(str - x); <- Führt zu Kompilierungsfehler
```

Statically vs. dynamically typed

- ▶ Sprachen wie Java
- ▶ Datentyp wird einmal deklariert und steht fest
- ▶ Nach `int x = 3;` sind `String x = "Hello";` und `x = "There";` nicht mehr möglich.
- ▶ Python: `x = 3` dann `x = "Ah General Kenobi"` möglich.

Compiled vs. interpreted

- ▶ Java kompiliert durch `javac` Befehl
- ▶ `.class` Datei ist Maschinencode aus `.java` Datei
- ▶ Interpretierte Sprachen: JavaScript, Python, ...
- ▶ Kompilierte Sprachen meistens schneller

Garbage collected vs. manuelle Speicherallokation

- ▶ Unterschied im Speichermanagement
- ▶ Manuelle Speicherallokation wird von Programmierer:in kontrolliert
- ▶ Garbage collector automatisiert Speicher management
- ▶ Manuelle Speicherallokation ist schneller und braucht weniger Speicher ist aber schwieriger

Programmiersprachen

"There are only two kinds of languages: the ones people complain about and the ones nobody uses." [Str13]

HTML & CSS

- ▶ Bilden Grundlagen der Darstellung von Webseiten
- ▶ Mithilfe von anderen Sprachen auch in der Anwendungsentwicklung genutzt
- ▶ HTML: markup language und keine richtige Programmiersprache



Source: [webd]
Content: *Css und HTML Logo*

JavaScript

- ▶ Steuer verhalten von Webseiten
- ▶ Eigenschaften: Interpretiert, dynamisch typisiert, keine Typsicherheit, garbage collected, mehrere Paradigmen
- ▶ Unerlässlich für dynamische Webseiten
- ▶ Verwendung: Web- und Applikationsentwicklung, Machine Learning, Data science, game development



Source: [webh]
Content: *JavaScript Logo*

TypeScript

- ▶ Superset von JavaScript
- ▶ Statisch Typisiert und Typsicher
- ▶ Verwendung eines Transpilers
- ▶ Verwendung: Kann fast überall verwendet werden, wo JavaScript verwendet wird



Source: [webs]

Content: *TypeScript Logo*

PHP

- ▶ Eigenschaften: Interpretiert, dynamisch typisiert, garbage collected, mehrere Paradigmen
- ▶ Serverseitiges rendern
- ▶ Verwendung: Webentwicklung
- ▶ Grundlage für Wordpress und Wikipedia



Source: [webl]
Content: *Php Logo*

Ruby

- ▶ Eigenschaften: Interpretiert, dynamisch typisiert, garbage collected, objektorientiert
- ▶ Verwendung:
 - ▶ Webentwicklung mit Ruby on Rails
 - ▶ Skripte auf Linux, Windows, iOS
 - ▶ Vereinzelt auch Spieleentwicklung



Source: [webo]
Content: *Ruby Logo*

Java und die JVM

- ▶ Eigenschaften: Kompiliert, statisch typisiert, typsicher, garbage collected, objektorientiert
- ▶ Plattformunabhängigkeit dank JVM
- ▶ Grundlage von Android Apps
- ▶ Verwendung: Web- und Applikationsentwicklung, data science, künstliche Intelligenz, Spieleentwicklung (Minecraft)
- ▶ JVM ermöglicht Nutzung in anderen Sprachen: Kotlin, Clojure, Scala



Source: [webg]
Content: Java Logo

Kotlin

- ▶ Basiert auf der JVM
- ▶ Kann Java code nutzen
- ▶ neue Funktionalitäten im Vergleich zu Java
- ▶ Verwendung:
 - ▶ Überall, wo Java verwendet wird
 - ▶ Beliebteste Sprache für Android Applikationsentwicklung



Source: [webi]
Content: *Kotlin Logo*

Dart

- ▶ Eigenschaften: Kompiliert, statisch oder dynamisch typisiert, garbage collected, objektorientiert
- ▶ Von google als JavaScript Nachfolger entwickelt
- ▶ Verschiedene Kompilierungsziele
- ▶ Verwendung:
 - ▶ Cross-platform Entwicklung mit Flutter
 - ▶ Noch nicht viel Verwendung in anderen Gebieten



Source: [webe]
Content: *Dart Logo*

Golang

- ▶ Eigenschaften: Kompiliert, statisch oder dynamisch typisiert, garbage collected, prozedural
- ▶ Benutzt von Google, Uber und Twitch
- ▶ Entwickelt von Google
- ▶ Verwendung:
 - ▶ Infrastruktur für Webanwendungen
 - ▶ Applikationsentwicklung
 - ▶ Künstliche Intelligenz
 - ▶ Eingebettete Systeme
 - ▶ Spieleentwicklung



Source: [webf]
Content: *Golang Logo*

Sprachen mit manueller Speicherallokation

- ▶ Drei viel benutzte Sprachen mit manuellem Speichermanagement
- ▶ C, C++, Rust
- ▶ Verwendung:
 - ▶ Eingebettete Systeme
 - ▶ Aber auch in anderen Bereichen verwendet
- ▶ Vorteil in Laufzeiten

C

- ▶ Älteste Sprache
- ▶ Nur Unterstützung von Prozeduraler Programmierung
- ▶ Grundlage für Sprachen wie: Java, Python, Objective-C, C++, C#
- ▶ Grundlage für die meisten Betriebssysteme



Source: [weba]
Content: *C Logo*

C++

- ▶ Superset von C mit wenigen Ausnahmen
- ▶ Unterstützung von objektorientierter Programmierung
- ▶ Verwendung:
 - ▶ Eingebettete Systeme
 - ▶ Web- und Applikationsentwicklung
 - ▶ Spieleentwicklung, Grundlage für Unreal engine



Source: [webb]
Content: C++ Logo

Rust

- ▶ Neuste von den Drei Programmiersprachen
- ▶ Integration von neuen Konzepten
- ▶ Große Beliebtheit



Source: [webp]
Content: *Rust Logo*



Source: [webq]
Content: *Rust Maskottchen "Feris"*

C# und das .NET Framework

- ▶ Ähnelt eher Java als C++
- ▶ Eigenschaften: Kompiliert, statisch typisiert, typsicher, garbage collected, objektorientiert
- ▶ Gestützt durch das .NET Framework
- ▶ Grundlage für viele Windows Applikationen
- ▶ Interaktion mit andere Sprachen durch .NET



Source: [webc]
Content: C# Logo

Python

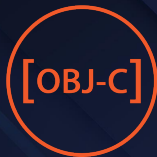
- ▶ Sehr beliebte Sprache
- ▶ Einfache Syntax
- ▶ Eigenschaften: Interpretiert, dynamisch typisiert, garbage collected, mehrere Paradigmen
- ▶ Verwendung in fast allen Bereichen
- ▶ Beliebteste Sprache für machine Learning
- ▶ Langsame Laufzeit



Source: [webm]
Content: *Python Logo*

Objective-C und Swift

- ▶ Apples Sprachen
- ▶ Kaum verwendet in anderen Bereichen



Source: [webk]
Content: *Inoffizielles Objective-C Logo*



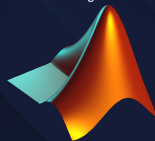
Source: [webr]
Content: *Swift Logo*

R & Matlab

- ▶ Domänen spezifische Sprachen
- ▶ R in Data science und Statistik
- ▶ Matlab für mathematische Probleme
- ▶ Kaum Verwendung außerhalb ihrer Domäne



Source: [webn]
Content: *R Logo*



Source: [webj]
Content: *Matlab Logo*

Zusammenfassung des Inhalts

- ▶ 19 Sprachen wurden vorgestellt
- ▶ Überblick verschafft
- ▶ Nicht ansatzweise Vollständig
- ▶ Sprachen aus der Vergangenheit aber auch aktuelle Sprachen wurden ausgelassen

Was solltet ihr mitnehmen?

- ▶ Jede Sprache hat Vor- und Nachteile
- ▶ Jede Sprache hat ihren Verwendungszweck
- ▶ Neues wird täglich erfunden
- ▶ Neue Sprachen können Erfahrungen Erweitern und eigenes Denken verbessern

Schlusswort

- ▶ Erstmal eine Sprache lernen
- ▶ Konzepte sind übertragbar
- ▶ Java ist gut für den Einstieg

References I



STROUSTRUP, Bjarne:
The C++ Programming Language.
4th.
Addison-Wesley Professional, 2013. –
ISBN 0321563840



C.
https://upload.wikimedia.org/wikipedia/commons/1/19/C_Logo.png, .
–
Last accessed 10 September 2022



C++.
https://upload.wikimedia.org/wikipedia/commons/thumb/1/18/ISO_C%2B%2B_Logo.svg/1822px-ISO_C%2B%2B_Logo.svg.png, . –
Last accessed 10 September 2022

References II



C#.

<https://banner2.cleanpng.com/20180831/iua/kisspng-c-programming-language-logo-microsoft-visual-stud-atlas-port-1956912415357423546294.jpg>, . -

Last accessed 10 September 2022



css & HTML.

https://upload.wikimedia.org/wikipedia/commons/thumb/1/10/CSS3_and_HTML5_logos_and_wordmarks.svg/1280px-CSS3_and_HTML5_logos_and_wordmarks.svg.png, . -

Last accessed 10 September 2022



dart.

<https://upload.wikimedia.org/wikipedia/commons/7/7e/Dart-logo.png>, . -

Last accessed 10 September 2022

References III



golang.

https://upload.wikimedia.org/wikipedia/commons/thumb/0/05/Go_Logo_Blue.svg/1200px-Go_Logo_Blue.svg.png, . –

Last accessed 10 September 2022



java.

<https://upload.wikimedia.org/wikipedia/de/thumb/e/e1/Java-Logo.svg/1200px-Java-Logo.svg.png>, . –

Last accessed 10 September 2022



js.

https://upload.wikimedia.org/wikipedia/commons/thumb/9/99/Unofficial_JavaScript_logo_2.svg/2048px-Unofficial_JavaScript_logo_2.svg.png, . –

Last accessed 10 September 2022

References IV



kotlin.

https://upload.wikimedia.org/wikipedia/commons/thumb/0/06/Kotlin_Icon.svg/2048px-Kotlin_Icon.svg.png, . –

Last accessed 10 September 2022



matlab.

https://upload.wikimedia.org/wikipedia/commons/thumb/2/21/Matlab_Logo.png/667px-Matlab_Logo.png, . –

Last accessed 10 September 2022



Objective-c.

<https://seeklogo.com/images/O/objective-c-logo-81746870EF-seeklogo.com.png>, . –

Last accessed 10 September 2022

References V



php.

https://upload.wikimedia.org/wikipedia/commons/thumb/3/31/Webysther_20160423_-_Elephpant.svg/2560px-Webysther_20160423_-_Elephpant.svg.png, . -

Last accessed 10 September 2022



Python.

<https://upload.wikimedia.org/wikipedia/commons/thumb/c/c3/Python-logo-notext.svg/1869px-Python-logo-notext.svg.png>, . -

Last accessed 10 September 2022



R.

https://upload.wikimedia.org/wikipedia/commons/thumb/1/1b/R_logo.svg/724px-R_logo.svg.png, . -

Last accessed 10 September 2022

References VI



ruby.

https://upload.wikimedia.org/wikipedia/commons/thumb/7/73/Ruby_logo.svg/1024px-Ruby_logo.svg.png, . –

Last accessed 10 September 2022



Rust.

https://upload.wikimedia.org/wikipedia/commons/thumb/d/d5/Rust_programming_language_black_logo.svg/1024px-Rust_programming_language_black_logo.svg.png, . –

Last accessed 10 September 2022



rust-mascot-feris.

<https://rustacean.net/assets/rustacean-flat-happy.png>, . –

Last accessed 10 September 2022

References VII



swift.

<https://developer.apple.com/swift/images/swift-og.png>, . –

Last accessed 10 September 2022



ts.

https://upload.wikimedia.org/wikipedia/commons/thumb/4/4c/Typescript_logo_2020.svg/1200px-Typescript_logo_2020.svg.png, . –

Last accessed 10 September 2022