

Einführung Java

Programmiermethodik

Lukas Kaltenbrunner, Simon Priller

Universität Innsbruck



Programmiersprachen

Programmierparadigma

- Programmierparadigma = Sichtweise auf und Umgang mit den zu verarbeitenden Daten und Operationen.
- Beispiele
 - Imperatives Programmierparadigma (*LV Einf. in die Programmierung*)
 - Folge von Anweisungen die streng sequenziell abgearbeitet wird.
 - Objektorientiertes Programmierparadigma (*diese LV*)
 - Verfügt über weitergehende Konzepte wie Klassen, Vererbung und Polymorphie.
 - Funktionales Programmierparadigma (*LV Funktionale Programmierung*)
 - Rein funktionale (applikative) Programmierung kennt keine Wertzuweisung.
 - Reihe von Funktionsaufrufen, die eine Eingabe in eine Ausgabe transformiert.
 - Logisches Programmierparadigma
 - Verwendet Logik zur Darstellung und Lösung von Problemen.
 - Programm besteht aus einer Menge von Axiomen, aus denen der Interpreter eine Lösungsaussage berechnet.

Abstraktionsgrad

- Der Abstraktionsgrad definiert, wie weit sich die Semantik der einzelnen Sprachkonstrukte von den Grundbefehlen des Prozessors unterscheidet.
- Abstraktionsgrade
 - Maschinensprache
 - Wird direkt vom Prozessor verstanden
 - Assembler
 - Einsatz symbolischer Namen, Sprungmarken etc.
 - Höhere Programmiersprachen
 - Kontrollstrukturen, Datentypen etc.
 - Objektorientierte Programmiersprachen
 - Daten und Methoden bilden eine Einheit.
 - Deklarative Programmiersprachen (Funktional/Logisch)
 - Kein Unterschied zwischen Daten und Operationen

Steigender Abstraktionsgrad

Ausführungsschema

Übersetzende Programmiersprachen (z.B. C, C++)

- Der Quelltext eines Programms wird vor der ersten Ausführung durch einen Übersetzer in eine Zielsprache (üblicherweise Maschinensprache) übersetzt.
- Hohe Ausführungsgeschwindigkeit
- Überprüfung auf Fehler

Interpretierende Programmiersprachen

- Der Quelltext eines Programms wird zur Laufzeit von einem Interpreter eingelesen und Befehl für Befehl abgearbeitet.
- Langsamer
- Flexibler (bei kleinen Änderungen)

Mischformen (z.B. Java)

- Programm wird in einen Zwischencode übersetzt.
- Übersetzter Code wird interpretiert (virtuelle Maschine).



Java – Eine kurze Einführung

Java (1)

- Start 1991
- Sun Microsystems (seit 2010 Tochterunternehmen der Oracle Corporation)
- Ursprünglicher Name “Oak” (Object Application Kernel)
 - Angelehnt an C++
 - Elemente aus Smalltalk (Bytecode, Garbage Collection)
- Ziel war die Entwicklung einer Hochsprache für hybride Systeme im Consumer-Electronic Bereich.
 - Kompakte Programme
 - Plattform-unabhängige Programme
- Neuausrichtung Projekt → auch für Internet-Programmierung geeignet
- Sun Demo Web-Browser HotJava in den 90-ern (Applets)
 - Kleine Programme als Applets in HTML-Seiten

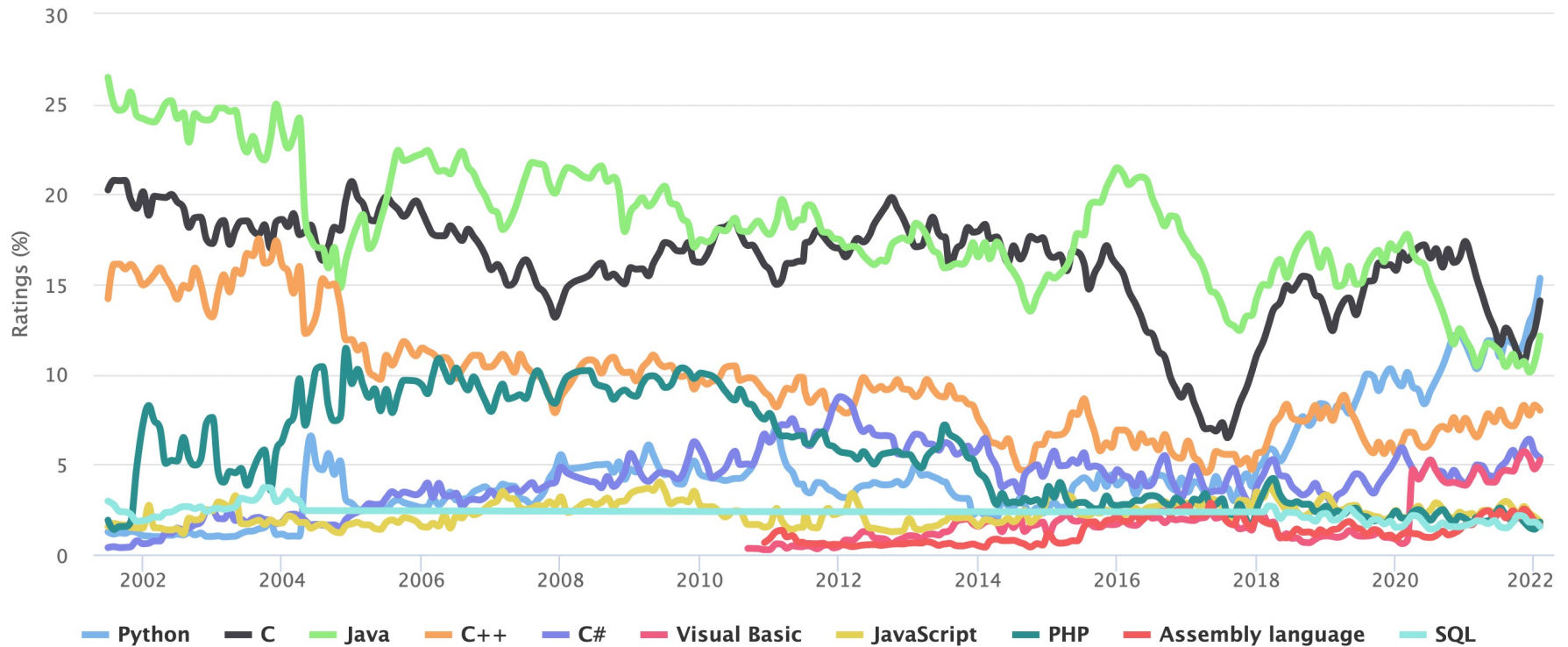
Java (2)

- Durchbruch 1995
 - Netscape Navigator 2.0 mit integrierter Java Virtual Machine (JVM)
- Aktuell
 - Java 13 Release September 2019
 - Java 14 Release März 2020
 - Java 15 Release September 2020
 - Java 16 Release März 2021
 - Java 17 Release September 2021
 - Java 18 geplanter Release März 2022
 - Release-Zyklus: 6 Monate (seit 2017)
- Java ist nicht nur reine Programmiersprache, bringt auch Plattformcharakter mit sich (über JVM).

Java (3)

TIOBE Programming Community Index

Source: www.tiobe.com



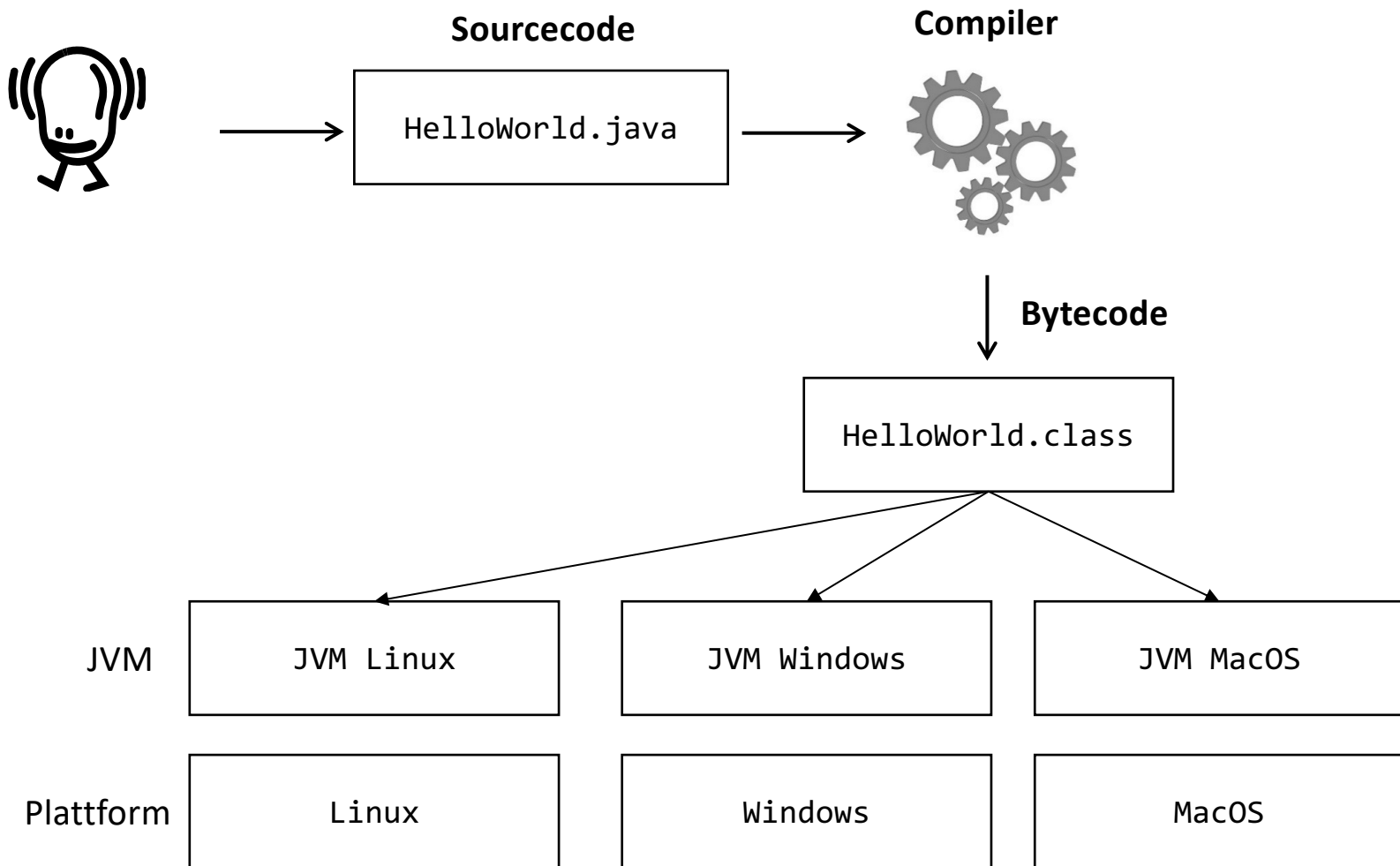
Eigenschaften der Sprache Java

- Portierbar
- Objektorientiert
- Robust & Sicher
 - Strenges Typsystem
 - Behandlung von Ausnahmen (Exception Handling)
 - Automatische Speicherbereinigung (Garbage Collection)
- Nebenläufig (Threading)

Erstellen/Ausführen von Programmen

- Java Compiler
 - Übersetzt den Programmtext in einen einfachen künstlichen Code (Bytecode)
 - Bytecode ist für alle Rechner/Betriebssysteme gleich.
- Java Virtual Machine (JVM)
 - "Virtueller, künstlicher" Computer
 - Dient als Schnittstelle zur Maschine und dem Betriebssystem.
 - Für jede Plattform existiert eine eigene JVM.
 - Liest den Bytecode und führt ihn aus (Interpretation!).
 - Mehr in der entsprechenden Vorlesung

Hybrider Ansatz in Java



Bytecode

- Plattform-unabhängig und meist sehr kompakt
→ *write once, run anywhere*
- Enthält keine Anweisungen in Maschinensprache für den physischen Computer.
- Interpreter für Bytecode ist kleiner und schneller als ein Interpreter für den ursprünglichen Source-Code.
- Endbenutzer*innen haben keinen Zugriff auf den Source-Code.
- Der Interpreter kann zusätzliche Überprüfungen durchführen (z.B. Arraygrenzen).

Programm in Java

- In der Einführung in die Programmierung wurden verschiedene Grundelemente von Programmiersprachen behandelt, z.B.:
 - Datentypen, Variablen
 - Anweisungen
 - Zeiger
 - Modularisierung
- In Java, einer objektorientierten Sprache, sind alle Programmteile einer Klasse zugeordnet.
- Jedes Programm hat zumindest eine Klassendefinition mit mindestens einem Unterprogramm (Methode).
- Es muss eine „Start“-Methode geben - in Java ist dies die `main`-Methode.

Hello World! (C und Java)

- HelloWorld.c

```
#include <stdio.h>
#include <stdlib.h>

int main(void) {
    printf("Hello World!\n");
    return EXIT_SUCCESS;
}
```

- HelloWorld.java

```
public class HelloWorld {

    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

Hello (Java) World!

```
public class HelloWorld {  
  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

- Eine Klasse wird durch das Schlüsselwort `class` vereinbart.
- Die Klasse erhält den Namen `HelloWorld`:
 - Der Code muss in einer Datei `HelloWorld.java` stehen.
 - Die durch das Kompilieren erhaltene Bytecode-Datei heißt `HelloWorld.class`.
- Die `main`-Methode ist der Startpunkt der Ausführung.
 - Die `main`-Methode muss immer `public` und `static` sein (wird später im Kapitel über Objektorientierung noch ausführlich erklärt).
 - Die `main`-Methode gibt nichts zurück, daher `void`.
 - Dem Programm (der `main`-Methode) können Parameter übergeben werden (wird im Kapitel über Arrays genauer erklärt).
- In der `main`-Methode können beliebige Anweisungen stehen.

Übersetzen und Ausführen

- Auf der Kommandozeile (wenn entsprechende Programme installiert sind)
 - Übersetzen
`javac HelloWorld.java`
 - Ausführen
`java HelloWorld`
- Entwicklungsumgebungen (Beispiele)
 - Eclipse: <http://www.eclipse.org/>
 - IntelliJ: <https://www.jetbrains.com/idea/>
 - Visual Studio Code: <https://code.visualstudio.com/>