

Teil 1: Schritt für Schritt zum ersten Java-Programm

Modul „Grundlagen der objektorientierten Programmierung
mit Java“

Prof. Dr. Cornelia Heinisch

Lernziele

- Sie kennen die Werkzeuge, die Sie benötigen, um
 - Java-Programme zu **erstellen**,
 - Java-Programme zu **übersetzen**,
 - Java-Programme **auszuführen**.
- Sie kennen die Begrifflichkeiten Rechner-Plattform, Java-Plattform, JDK und JRE.
- Sie haben Ihr erstes Java-Programm zur Ausführung gebracht.
- Sie kennen den Unterschied zwischen „übersetzten“ und „interpretierten“ Sprachen.

Schritt für Schritt zum ersten Java-Programm

Agenda

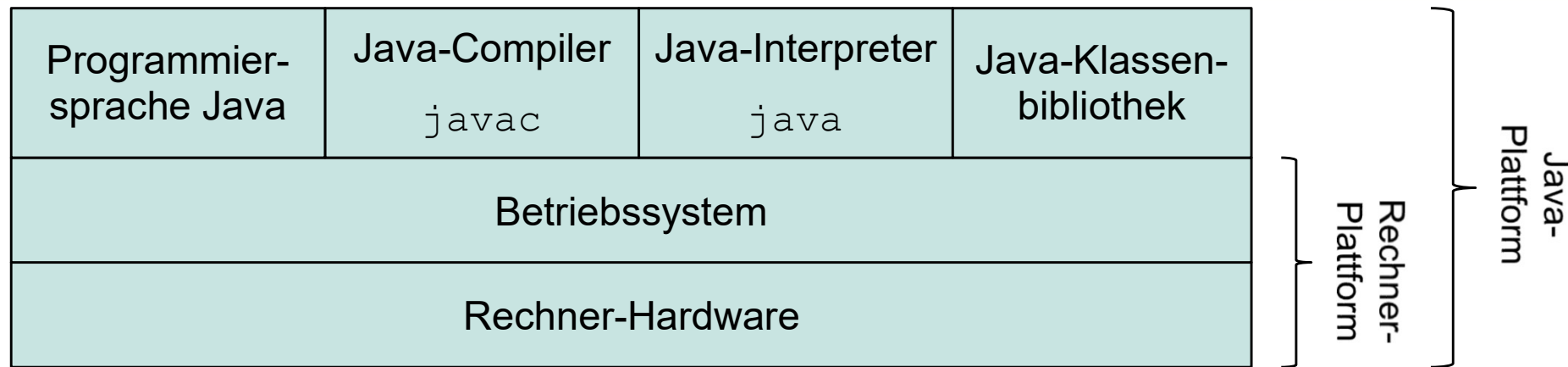
- Werkzeuge für die Programmierung in Java
- Das Programm „Hello, World!“
- Übersetzungskonzepte

Java-Programme schreiben und ausführen

Zum Ausführen und Schreiben von Java-Programmen benötigt man Folgendes:

- einen **Editor**,
um ein Java-Programm einzutippen – bzw. den **Java-Quellcode** zu erstellen,
- einen **Java-Compiler**,
um den im Editor eingetippten Java-Quellcode in **Java-Bytecode** zu übersetzen,
- die **Java-Klassenbibliothek**,
welche ausprogrammierte Funktionalitäten zur Verwendung in den eigenen Java-Programmen zur Verfügung stellt,
- einen **Java-Interpreter**,
um den Java-Bytecode auf der jeweiligen **Rechner-Plattform** auszuführen. Hierzu übersetzt der Java-Interpreter den Java-Bytecode in Maschinencode und bringt diesen direkt zur Ausführung.

Rechner-Plattform und Java-Plattform



Als **Rechner-Plattform** wird hier die Kombination von **Betriebssystem** und zugehöriger **Rechner-Hardware** verstanden.



Zu einer **Java-Plattform** gehören neben der **Programmiersprache Java**, Werkzeuge wie der **Java-Compiler** (`javac`), die **Java Virtuelle Maschine** (JVM) – in anderen Worten der **Java-Bytecode-Interpreter** (`java`) – für eine **Rechner-Plattform** und eine **umfassende Klassenbibliothek**.

Ausführungen der Java-Plattform

- Java-Plattform wird von Oracle in verschiedenen Ausführungen bereit gestellt:
 - **Standard Edition** (Java SE),
 - Enterprise Edition (Java EE),
 - Micro Edition (Java ME).
- Ausführungen unterscheiden sich im Wesentlichen durch
 - Art und Umfang der Klassenbibliothek,
 - die bereit gestellten Werkzeuge.
- Hintergrund für unterschiedliche Ausführungen:
Einsatz von Java-Programmen auf unterschiedlichen Endgeräten.



Sie benötigen für dieses Modul eine Java-Plattform in der Standard Edition von der Firma Oracle.

Java Development Kit (JDK)

- meist genutzte Java-Plattform für die Entwicklung von Java-Programmen
- wird von der Firma Oracle zur Verfügung gestellt (Open JDK):
<https://jdk.java.net/20/>



Das JDK ist eine Java-Plattform mit einem für die jeweilige Rechner-Plattform spezifischen Java-Bytecode-Interpreter.

- Für die Rechner-Plattform entsprechendes Build auswählen.

Java Development Kit (JDK) – Hinweise zur Installation

- Zip-Datei herunterladen
- Entpacken zum Beispiel in Verzeichnis `C:\Programme\Java`
- Nach dem Entpacken finden Sie dort das Verzeichnis
 - `jdk-20`
In diesem Verzeichnis finden sich unter anderem folgende Verzeichnisse:
 - `\bin`: hier befinden sich die Werkzeuge `javac.exe` (Java-Compiler) und `java.exe` (Java-Interpreter)
 - `\lib`: hier befindet sich auf mehrere Dateien aufgeteilt die Java-Klassenbibliothek

Unterscheidung JRE und JDK

- JRE = Java Runtime Environment
- JRE beinhaltet nur diejenigen Bestandteile eines JDKs, welche zum Ausführen von Java-Programmen benötigt werden.
- damit besteht eine JRE aus
 - einem Bytecode-Interpreter für die jeweilige Rechner-Plattform
 - und der Java-Klassenbibliothek.

Entwicklungsumgebung Eclipse

- ist selbst vollständig in Java geschrieben
- bietet über komfortable grafische Bedienoberfläche einfachen und direkten Zugriff auf alle benötigten Entwicklungswerkzeuge:
 - komfortabler Editor, um Java-Programme einzutippen,
 - Syntax des Programmes wird bereits beim Eintippen überprüft,
 - beim Speichern wird automatisch der Java-Compiler aufgerufen,
 - Bedienelemente, um Java-Programme direkt auszuführen oder zu debuggen.
- Bezugsquelle: <http://www.eclipse.org/downloads/>
 - Installationsdatei starten (als Administrator ausführen)
 - „Eclipse IDE for Java-Developers“ auswählen

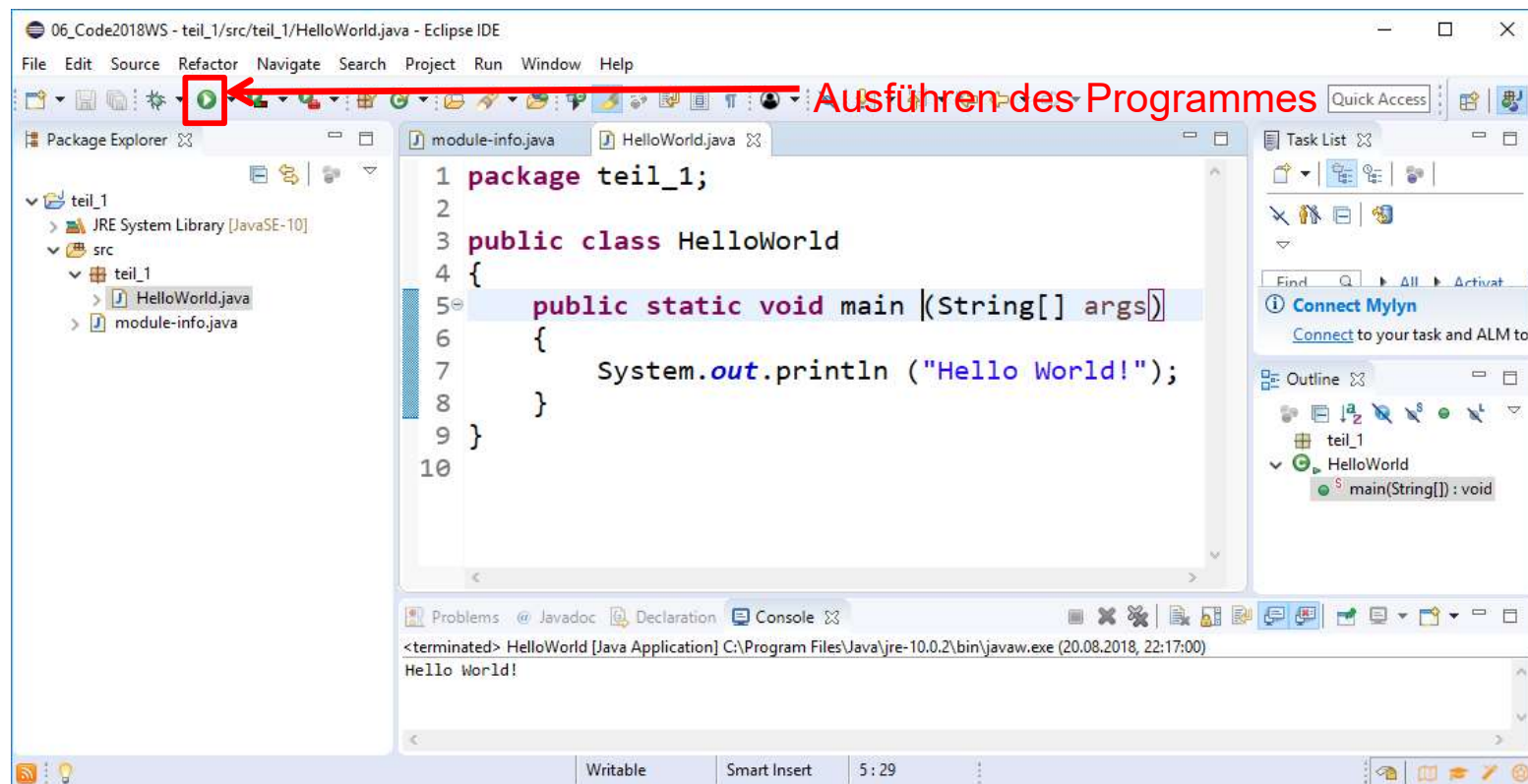
Schritt für Schritt zum ersten Java-Programm

Agenda

- Werkzeuge für die Programmierung in Java
- Das Programm „Hello, World!“
- Übersetzungskonzepte

Entwicklungsumgebung Eclipse

- grafische Bedienoberfläche von Eclipse nach Eintippen & Speichern eines Beispielprogrammes
- unterhalb des Editor-Fensters ist nach dem Ausführen die Programmausgabe zu sehen



Schritt für Schritt zum ersten Java-Programm

Agenda

- Werkzeuge für die Programmierung in Java
- Das Programm „Hello, World!“
- Übersetzungskonzepte

Analogie



Was ist der Unterschied zwischen einem Übersetzer und einem Dolmetscher?

Person A



schreibt

Brief in
Sprache A

Hallo Herr
Peter,

Übersetzer



liest übersetzt schreibt

Brief in
Sprache B

Hello Mr.
Peter,

Person B



liest

Person A



sagt

Sprache A

Satz 1
Satz 2
Satz 3
.....

Dolmetscher



hört übersetzt sagt

Sprache B

Satz 1
Satz 2
Satz 3
.....

Person B

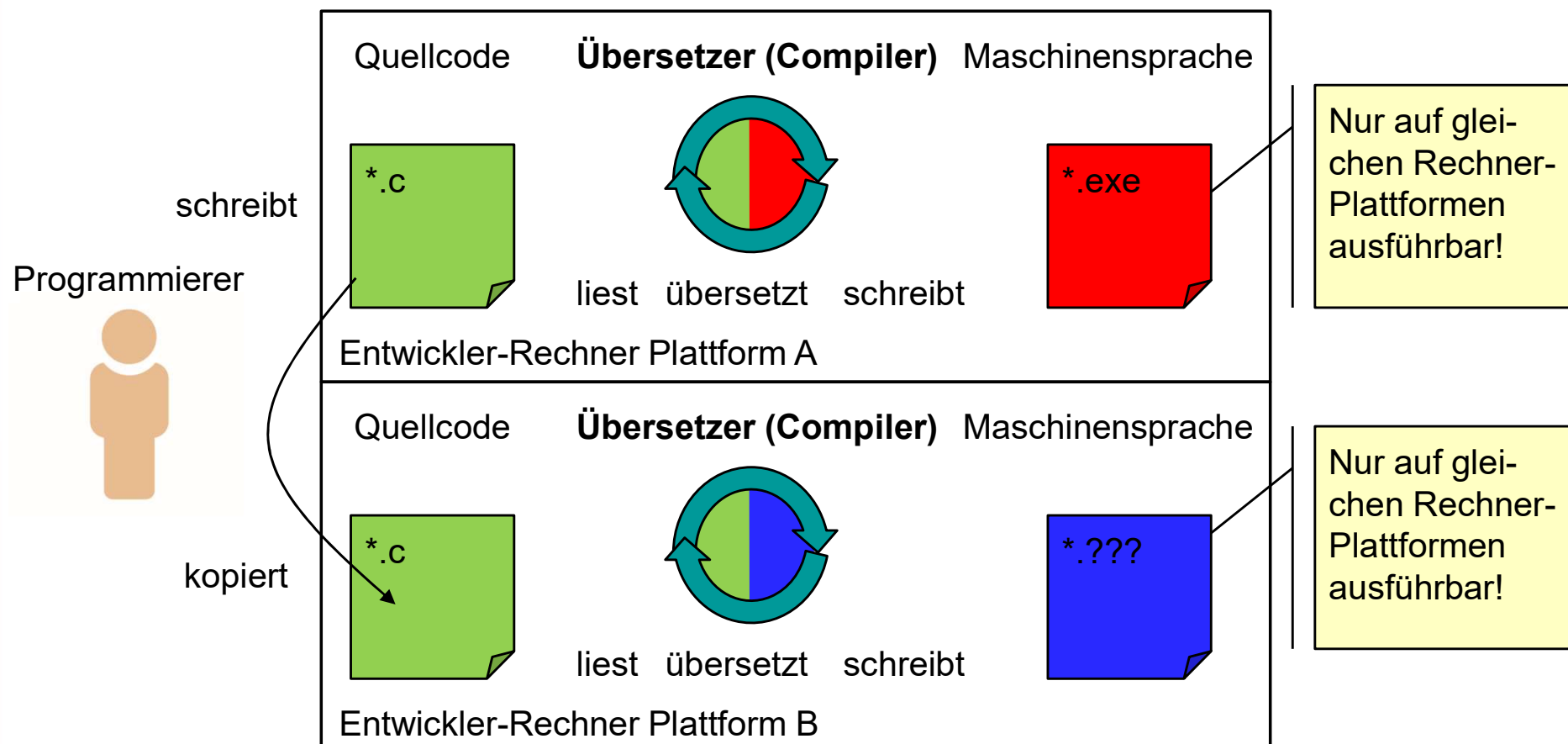


hört

Programmiersprache C



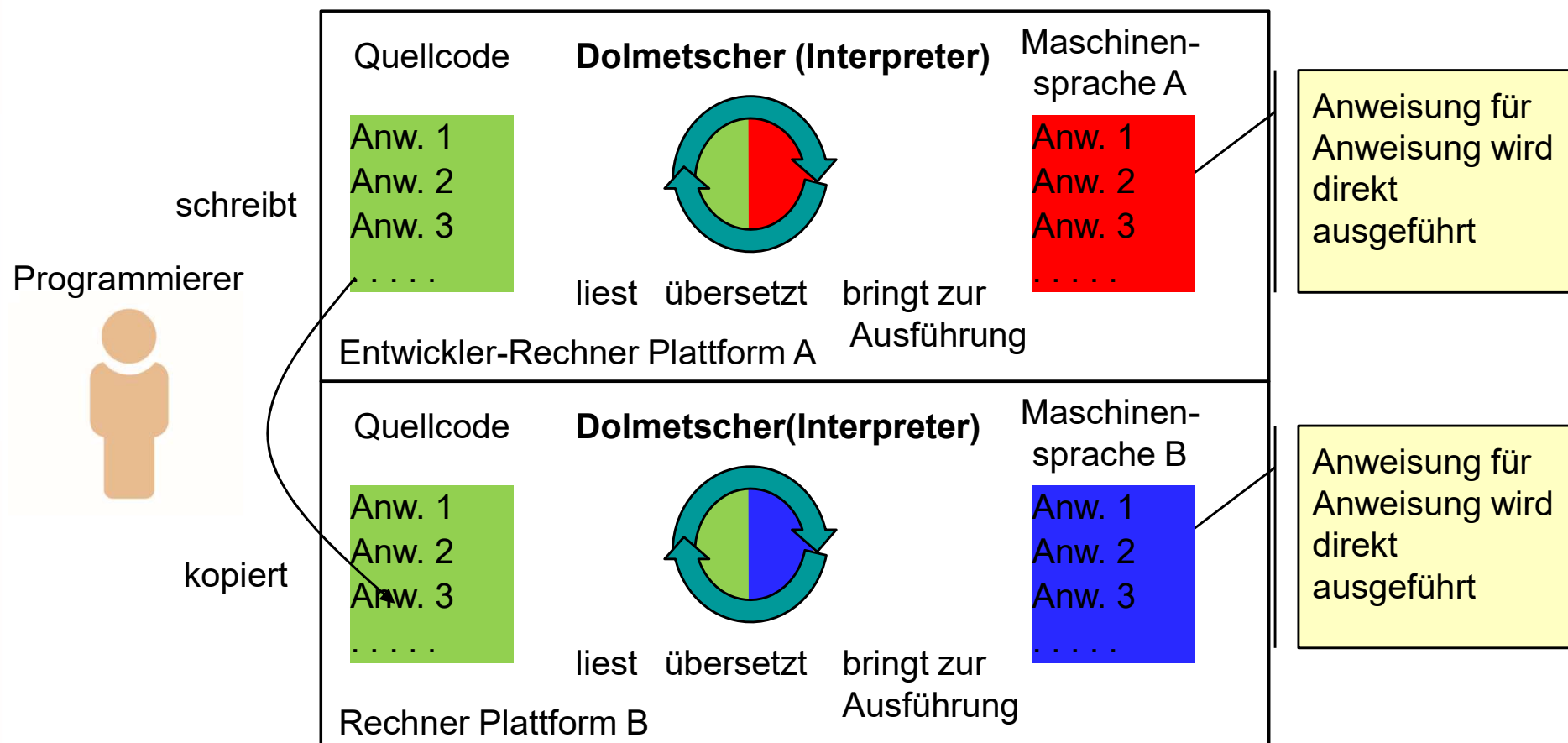
Quellcode wird in Maschinensprache übersetzt. Für jeden Maschinentyp muss eine ausführbare Datei erstellt werden.



Rein interpretierte Sprachen



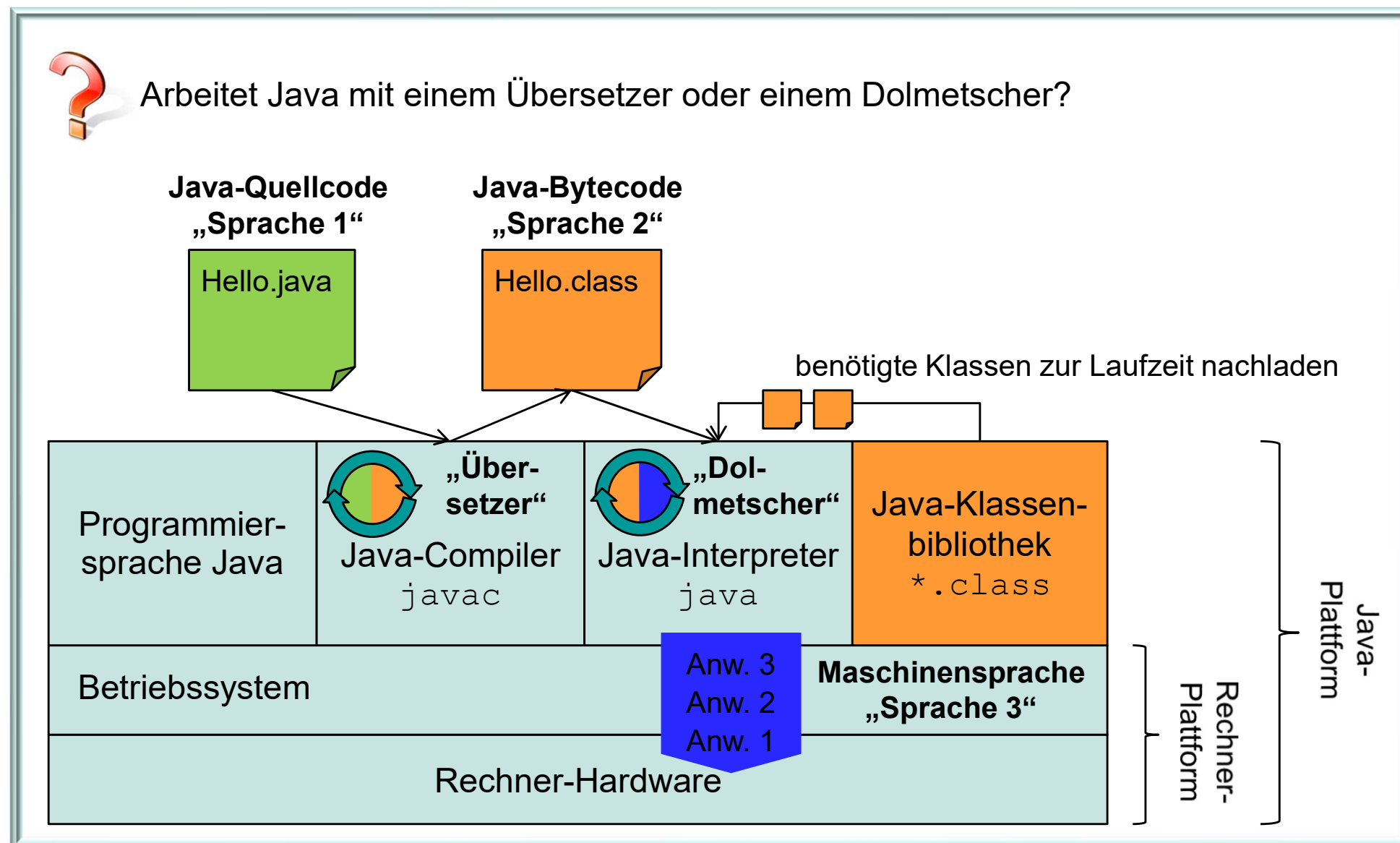
Quellcode wird Anweisung für Anweisung in Maschinsprache übersetzt und direkt ausgeführt. Es gibt keine ausführbare Datei!



Programmiersprache Java



Arbeitet Java mit einem Übersetzer oder einem Dolmetscher?



Zweistufiges Übersetzungskonzept



Warum arbeitet Java mit einem Zwischencode?

- Rein interpretierte Sprachen sind langsamer in der Ausführung als rein übersetzte Sprachen. Java geht den Kompromissweg: Großteil der Übersetzungsarbeit ist bereits erledigt. **Bytecode** (Zwischencode) **ist** bereits **Hardware-nah**.
- **Bytecode ist kompakter** als der ursprüngliche Quellcode. Damit sind die kleinen Bytecode-Dateien effizient über das Internet übertragbar.

In Java

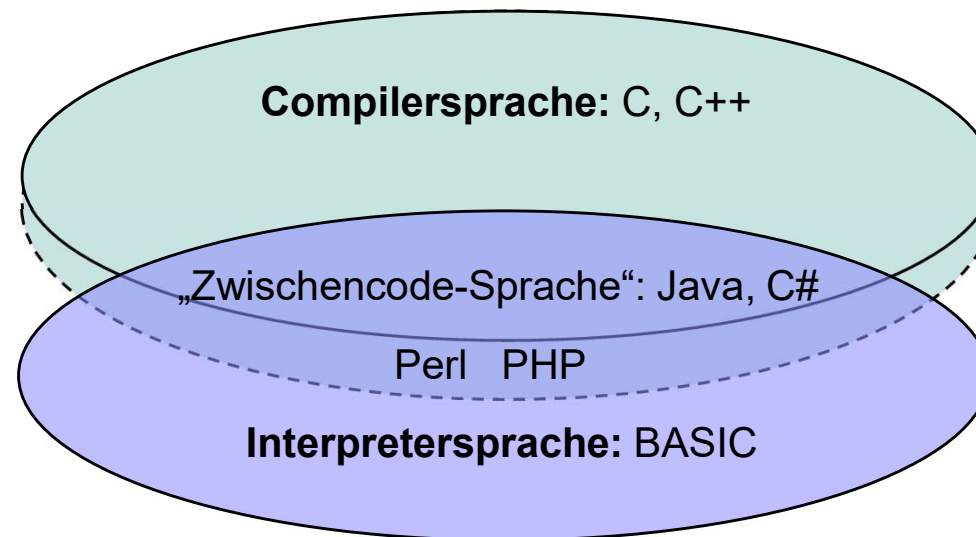
- gibt es damit kein ausführbares Programm.
- die Übersetzung des Bytecodes in Maschinencode erfolgt erst zur Laufzeit



Bytecode kann auf jeder Rechner-Plattform ausgeführt werden, sofern dort ein Dolmetscher (Java-Runtime-Environment) installiert ist.

Typische Compilersprachen und Interpretersprachen

- Rein übersetzte Sprache -> Compilersprache
- Rein interpretierte Sprache -> Interpretersprache
- Interpretersprachen werden auch häufig Skriptsprachen genannt
- Viele Skriptsprachen sind aber keine reinen Interpretersprachen
- Beispiele:



Eine Programmiersprache ist prinzipiell unabhängig vom Übersetzungskonzept. Typischerweise ist C aber eine Compilersprache und Java eine Sprache, die mit Compiler, Bytecode und Interpreter arbeitet.