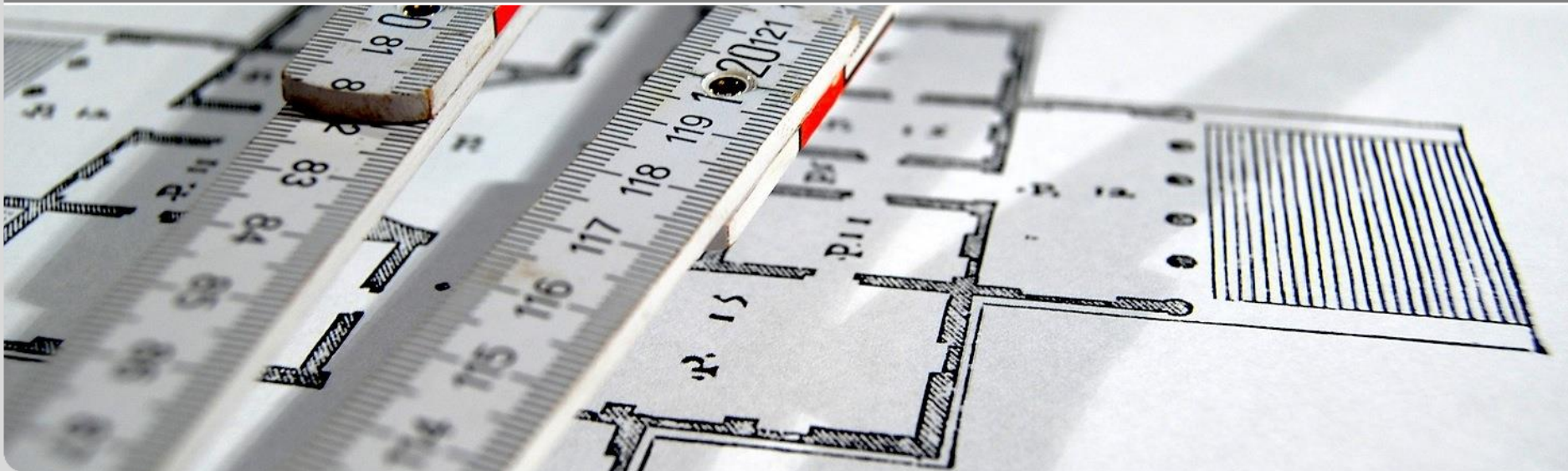


Programmieren-Tutorium Nr. 10

1. Tutorium | Jonas Ludwig
Java: Installieren, Kompilieren und Ausführen

Architecture-driven Requirements Engineering – Institut für Programmstrukturen und Datenorganisation – Fakultät für Informatik



Wer bin ich?

- Jonas Ludwig
 - Informatik Bachelor im 5. Semester
- Mail: jonas.ludwig@student.kit.edu
- Folien werden zu jedem Tutorium im ILIAS hochgeladen
 - Magazin > Organisationseinheiten > Fakultät für Informatik > WS 19/20 > Programmieren > Tutorien > Tutorium 10 (Jonas Ludwig)

Wer seid ihr?

- Euer Name?
- Euer Studiengang?
- Euer Betriebssystem?
- Eure Programmierkenntnisse?
- Eure Erwartungen?
- Eure Wünsche?

Was machen wir Heute?

- Organisatorisches
 - Allgemeines
 - Termine
 - Übungsbetrieb

- Entwicklungsumgebung
 - Java Grundlagen
 - Java einrichten
 - Java ausführen

Das Tutorium Nummer 10

- Jeden Dienstag um 09:45 Uhr im Geb. 50.34, Raum -107
- Was werden wir im Tutorium machen?
 - Den Vorlesungsstoff wiederholen und vertiefen
 - Fragen zu den Übungsblättern besprechen
 - Fragen zu der Korrektur besprechen
 - Gemeinsam Programmieren mit Java lernen
 - Laptops sind durchaus erwünscht!
- Z-Folien für zusätzliche Informationen
 - Nicht wirklich relevant für „Programmieren“
 - Jedoch relevant für zukünftige Vorlesungen

Das Modul Programmieren

- Teil der Orientierungsprüfung
 - Muss bis zum 2. versucht und bis zum 3. Semester bestanden werden
- Übungsschein
 - 5 Übungsblätter + 1 Präsenzübung
 - Bestanden beim Erreichen von:
 - > 50% der Punkte in den Übungsblättern und
 - > 75% der Punkte in der Präsenzübung
 - Anmeldung im Campus Management Portal: **02.12.2019 - 08.01.2020**
- Abschlussaufgaben
 - 2 getrennte Aufgaben während der vorlesungsfreien Zeit
 - 100% der Abschlussnote
 - Übungsschein wird zur Anmeldung und Teilnahme vorausgesetzt
 - Anmeldung im Campus Management Portal: **11.02.2020 – 20.02.2020**

Die Übungsblätter

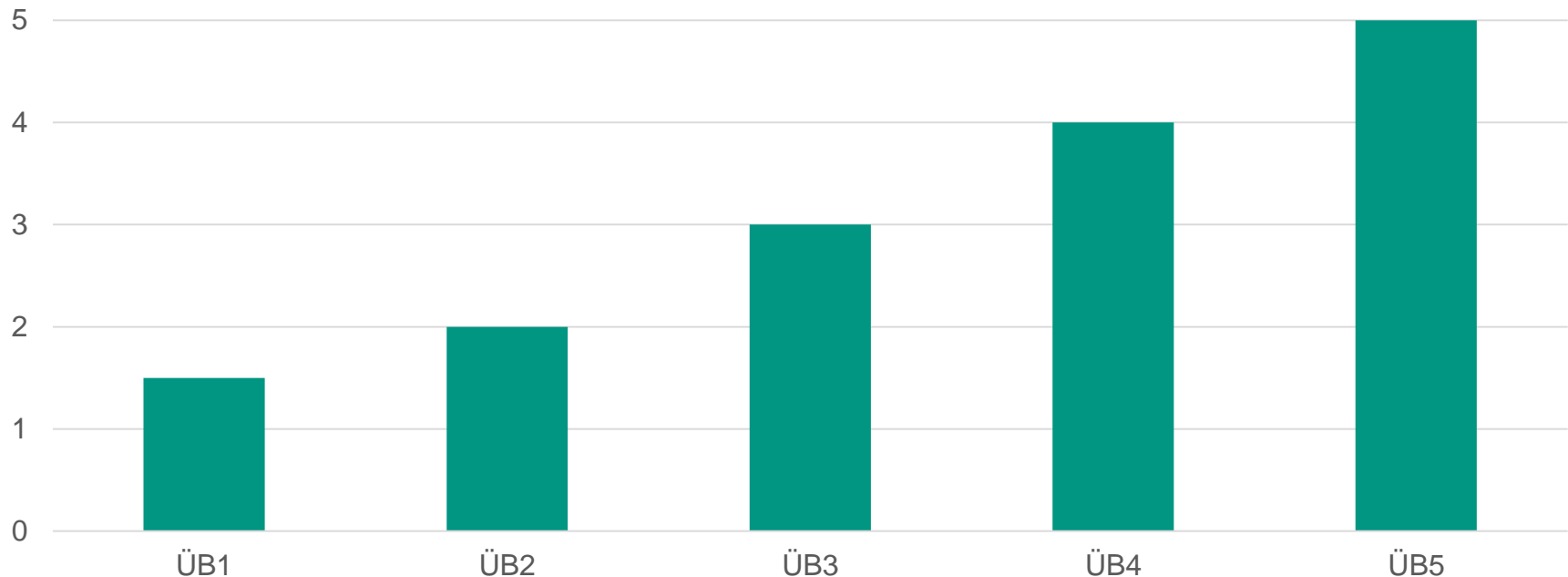
- Alle zwei Wochen ein neues Übungsblatt
 - Zwei Wochen Zeit zur Bearbeitung
 - Abgabefrist steht auf dem Übungsblatt
- Fünf Übungsblätter insgesamt
 - 4. Übungsblatt über die Weihnachtszeit
 - Vier Wochen Bearbeitungszeit
- Pro Blatt immer 20 Punkte
 - Insgesamt 100 Punkte
 - Mindestens benötigt 50,5 Punkte
- Das erste Blatt wird am 06. November veröffentlicht
 - Die Abgabe wird erst am 13. November um 13:00 Uhr freigeschaltet
 - Abgabe bis zum 21. November um 06:00 Uhr im Praktomat

Tipps für die Übungsblätter

- Die Übungsblätter werden schnell deutlich anspruchsvoller
 - Sammelt auf den ersten Blättern möglichst viele Punkte
- Nutzt die zwei vollen Wochen und nicht nur das letzte Wochenende
 - Unklarheiten können so frühzeitig beseitigt werden
- Nicht abschreiben!
 - Schon bei einmaligem Nachweis verwirkt man die Chance auf den Schein
 - Ohne Übungsschein darf man die Abschlussaufgabe nicht schreiben
 - Nächster Versuch erst wieder im 2. Semester
 - Programmieren ist eine Orientierungsprüfung!
- Weitere Informationen in der Prüfungsordnung:
 - <https://www.informatik.kit.edu/formulare.php>

Tipps für die Übungsblätter

Relative Schwierigkeit der Übungsblätter



- Angaben ohne Gewähr!
- Besonders für Anfänger **wichtig**:
 - Die ersten Übungsblätter **nicht vernachlässigen!**

Die Präsenzübung

- Am **15. Januar 2020** um **17:30 Uhr**
- Alle Aufgaben sind ohne jegliche Hilfsmittel zu lösen
- $\text{Übungsblätter} > 50\% \wedge \text{Präsenzübung} > 75\% \Rightarrow \text{Übungsschein}$
- $\text{Übungsblätter} \leq 50\% \vee \text{Präsenzübung} \leq 75\% \Rightarrow \text{Beides wiederholen}$
- $\text{Übungsblätter} > 50\% \wedge \text{ärztliches Attest} \Rightarrow \text{Präsenzübung wiederholen}$
- Machbar, wenn Übungsblätter ordentlich bearbeitet

Die Abschlussaufgaben

- Anmeldung erneut im Campus Management Portal
 - Die Anmeldung muss bis spätestens **20.02.2020 12:00** erfolgen
 - Erst mit eingetragensem Übungsschein möglich!
 - Erst nach der Korrektur des letzten Übungsblatt wird angefangen die Übungsscheine einzutragen

- Abgabe erfolgt – wie bei den Übungsblättern – über den Praktomat

- Abschlussaufgabe 1
 - Ausgabe: 10.02.2020 (ca. 13:00 Uhr)
 - Abgabe: 10.03.2020 (06:00 Uhr)

- Abschlussaufgabe 2
 - Ausgabe: 24.02.2020 (ca. 13:00 Uhr)
 - Abgabe: 24.03.2020 (06:00 Uhr)

Abschlussaufgaben der letzten Semester

■ SS 15

- Scrabble mit arithmetischen Ausdrücken
- Graphen-basiertes Fluginformationssystem

■ WS 15/16

- Navigationssystem mit Berechnung des kürzesten Pfades
- Brettspiel mit einem unendlichen Torusspielfeld

■ SS 16

- Mensch ärgere Dich nicht mit optionalen Regeln
- Studierendenportal zur Verwaltung und Berechnung von Noten

Abschlussaufgaben der letzten Semester

- WS 16/17
 - Literaturverzeichnis: Verwaltung von Schriftstücken
 - 3x3 Candy-Crush

- SS 17
 - Brettspiel: Ablaufen eines Feldes
 - Zustandstracker mit Historie und einfacher Auswertung

- WS 17/18
 - Brettspiel ähnlich „4 Gewinnt“ mit Torusspielfeld
 - Verwaltungssystem für Olympische Spiele

Vorlesungs-Homepage

- **<https://sdqweb.ipd.kit.edu/wiki/Programmieren>**
 - Eintrag im Wiki des Lehrstuhls „Software-Entwurf und -Qualität“

- Inhalte für das Modul Programmieren
 - Aktuelles
 - Vorlesungsfolien
 - Übungsblätter
 - Abschlussaufgaben
 - Links
 - Termine und Fristen
 - FAQ

- Das regelmäßige Durchlesen der Neuigkeiten wird vorausgesetzt!

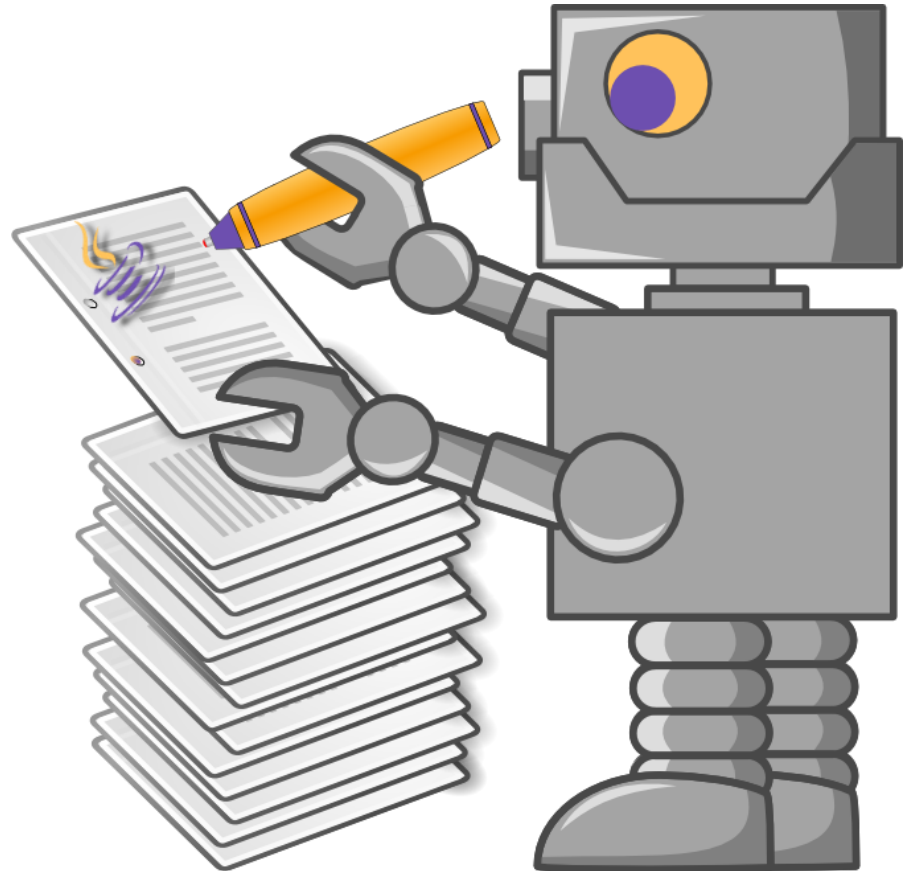
- https://ilias.studium.kit.edu/goto.php?target=crs_996576
 - Integriertes Lern-, Informations- und Arbeitskooperations-System
- Inhalte für das Modul Programmieren
 - Diskussionsforum
 - Vorlesung
 - Übungsaufgaben
 - Allgemein
 - Programmieren-Wiki
 - VPN
 - Java
 - Und vieles mehr
- Fragen zum Vorlesungsinhalt und den Übungsblättern ausschließlich in den ILIAS-Foren!

Einverständniserklärung - Disclaimer

- **<https://sdqweb.ipd.kit.edu/disclaimer/>**
 - Personalisierte Einverständniserklärung erzeugen
 - Drucken und Unterschreiben
- Abgabe der Einverständniserklärung bis spätestens **04.11.2019 12:00**
- Briefkasten für Programmieren hier im Untergeschoss
- Für Schülerstudenten, Anschrift und Geburtsdatum an:
programmieren-vorlesung@ipd.kit.edu

Praktomat

- Elektronisches Abgabesystem für die Übungsblätter
- Annotation der Quelltexte durch Tutoren
- Punkte-Übersicht für Tutanden
- Automatisierte Tests
 - Öffentlich und geheim
 - Verbindlich und unverbindlich
- Automatisierte Plagiatsprüfung
 - Nicht Abschreiben!

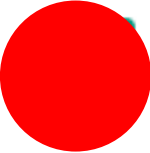


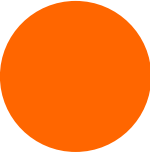
http://pp.ipd.kit.edu/projects/praktomat/Praktomat_Logo_stift.png


Praktomat

- Lösungen können beliebig häufig hochgeladen werden
 - Tutor sieht nur die letzte gültige Abgabe, sofern eine vorhanden ist!

- Ladet Teillösungen hoch
 - Sicherungskopie für euch
 - Eine vergessene Deadline ist nicht ganz so ärgerlich

-  Ungültige Abgabe mit fehlgeschlagenen verbindlichen Tests.
Tutor sieht diese nicht und kann auch keine Punkte verteilen!

-  Gültige Abgabe aber mit fehlgeschlagenem unverbindlichem Test

-  Gültige Abgabe und alle öffentlichen Tests bestanden

Praktomat

- https://praktomat.cs.kit.edu/2019_WS/
- Onlineanmeldung beim Praktomat bis spätestens **04.11.2018 12:00**
- Der Praktomat ist Open-Source
 - <https://github.com/KITPraktomatTeam/Praktomat/>
- Nur über das interne Uninetz erreichbar
 - ATIS oder SCC
 - WLAN am KIT (KIT, nicht eduroam!)
 - Remote Access (VPN)

Remote Access (VPN)

- <https://www.scc.kit.edu/dienste/vpn.php>
- Dieser Dienst stellt dem Anwender einen Zugang von extern zu internen Ressourcen des KIT zur Verfügung
- VPN-Client: OpenVPN
- Frühzeitig vor der ersten Abgabe einrichten und testen!
- FAQ: <http://www.scc.kit.edu/dienste/vpn-faq.php>
- Bei Fragen, Anregungen oder Problemen: vpn@scc.kit.edu

OpenVPN

- <http://www.scc.kit.edu/dienste/openvpn.php>

- Einrichtung
- Verwendung

- Verfügbar für:

- Windows
- Linux
- macOS
- Android
- iOS



Campus Management Portal

- <https://campus.studium.kit.edu>
- Funktionen des Campus Management Portal für Studierende
 - Studienbescheinigungen
 - Notenspiegel
 - Rückmeldung
 - Vorlesungsverzeichnis
 - Stundenplan (iCal)
 - Anmeldungen zu Veranstaltungen
- Über den Strukturbaum des Studiengangs kann sich zum Übungsschein und der Abschlusssaufgabe angemeldet werden

Informatik Studiengangservice

- <http://www.informatik.kit.edu/iss.php>
 - **Verbindliche** Antworten auf Ihre individuellen Fragen und Probleme
- Für Informatiker: beratung-informatik@informatik.kit.edu
- Für Informationswirtschaftler: beratung-inwi@informatik.kit.edu
- Für die **Sprechstunde** im Foyer des EG (bei der Info-Bib) in die Warteliste eintragen
- Sprechzeiten sind
 - Dienstags, 10:00 bis 12:00
 - Donnerstags, 10:00 bis 12:00

PROGRAMMIEREN IST ÜBUNGSSACHE!

- VL und Tut zwar sinnvoll, aber für Anfänger **unzureichend**
- **Einzig**e Möglichkeit, Programmieren *wirklich* zu lernen:
Programmieren! Was das Zeug hält!
- Ohne Eigeninitiative: Sehr wahrscheinlich Schwierigkeiten in späteren Semestern (SWT, Algorithmen, PSE, Propa,...)

Was tun bei Fragen?

- FAQ auf Vorlesungshomepage prüfen
- Im Wiki im ILIAS nachschauen
- Diskussionsforum prüfen ob Frage schon gestellt
- Im Diskussionsforum fragen
- Literatur (Uni-Bib/Internet)
- Tutor (also mich) fragen
- (Email an programmieren-vorlesung@ipd.kit.edu)



Literatur-Empfehlung: **Head First Java**

Literatur-Empfehlung: **Java ist auch eine Insel**

Online: <http://openbook.rheinwerk-verlag.de/javainsel/>

Was ist Java?

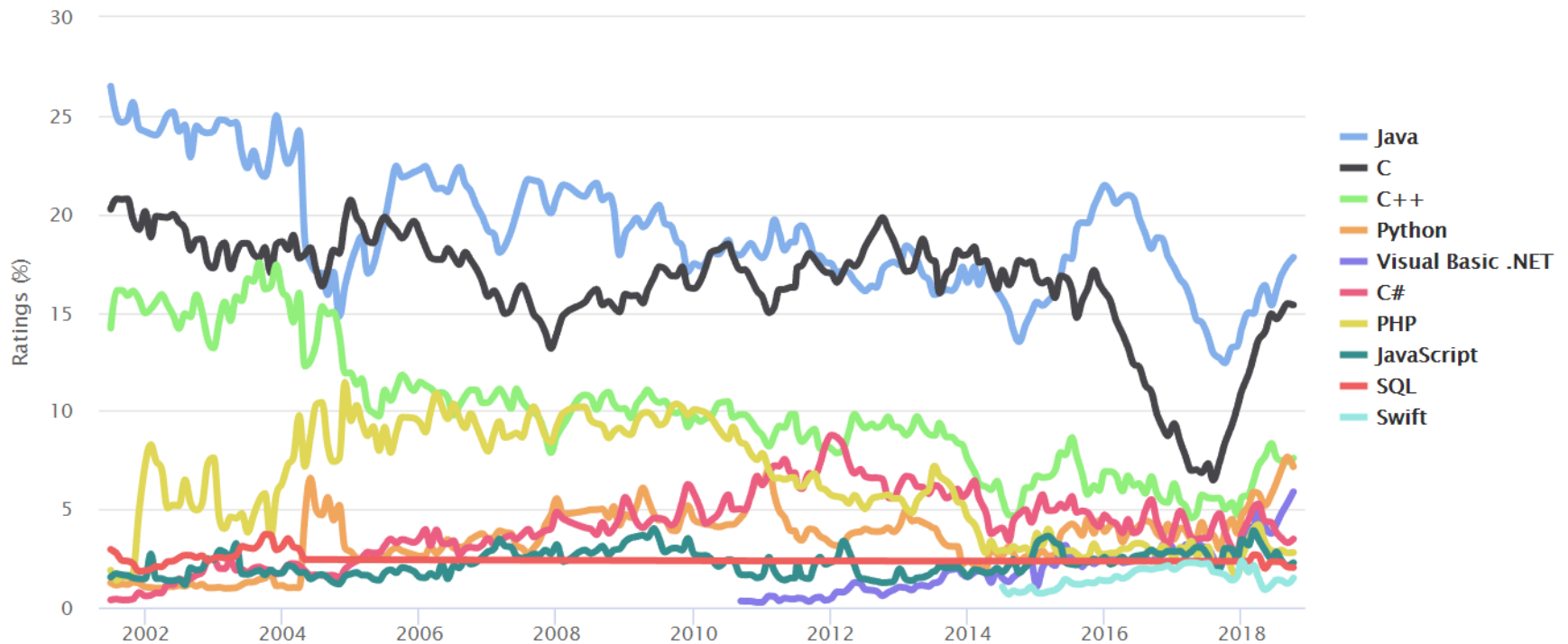
- Java ist eine *objektorientierte* Programmiersprache, welche auf der gleichnamigen Softwaretechnik aufbaut
- Java-Technologie, eine Sammlung von Softwarespezifikationen
 - Java-Plattform
- Programmiersprache: Formale Sprache zur Formulierung Rechenvorschriften, die von einem Computer ausgeführt werden
- Programmiersprachen haben eine eindeutig definierte:
 - Syntax – Welche Zeichenfolgen sind erlaubt
 - Semantik – Was bewirken die Zeichenfolgen auf dem Rechner



Warum Java?

TIOBE Programming Community Index

Source: www.tiobe.com



- Aktuell: Java 13
- Der Praktomat verwendet aber einen **Java 8 SE Compiler**
- Kompatibilität
 - Mit Java 13 kompilierbar Quelltext sollte auch mit Java 8 funktionieren
 - Aber: Bei der Verwendung von Java 13 spezifischen Features ist der Quelltext nicht mit dem Java 8 Compiler übersetzbar
- Verwendet also am besten auch Java 8

Java SE, Java EE oder Java ME?

- Die drei Editionen stellen neben verschiedene Bibliotheken (API) auch eine eigene virtuelle Maschine (Java VM) zur Verfügung
- **Java Platform, Standard Edition**
 - **Der Standard, welcher in der Vorlesung Verwendung findet**
- Java Platform, Enterprise Edition
 - Zur Entwicklung von Anwendungen für Unternehmen
- Java Platform, Micro Edition
 - Zur Entwicklung von Anwendungen auf mobilen Geräten

JRE oder JDK?

- Java Runtime Environment (JRE)
 - Laufzeitumgebung (`java`) um Java-Anwendungen weitgehend unabhängig vom darunter liegenden Betriebssystem auszuführen
- Java Development Kit (JDK)
 - Sammlung von Werkzeugen, um Java-Anwendungen zu erstellen
 - Beinhaltet neben der JRE und dem Java Compiler (`javac`) noch weitere Entwicklungswerkzeuge
- Weitere Informationen auf:
 - <https://docs.oracle.com/javase/8/docs/>

JDK unter Windows

- Java SE Development Kit 8
 - <https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>
- 32- oder 64-Bit?
 - <https://support.microsoft.com/kb/827218/de>
 - Explorer > Dieser PC (Rechtsklick) > Eigenschaften
- Die PATH-Systemvariable eintragen
 - <https://www.java.com/de/download/help/path.xml>
 - In Konsole testen “java –version”

OpenJDK unter Linux

- Bei Debian basierende Software-Distributionen, je nach Paketverwaltung, in das Terminal eingeben:
 - `sudo apt-get install openjdk-8-jre icedtea-8-plugin openjdk-8-jdk openjdk-8-source openjdk-8-demo openjdk-8-doc openjdk-8-jre-headless`
 - `sudo aptitude install ...`
- Bei Arch Linux:
 - `pacman -S jre8-openjdk`
- Weitere Informationen auf:
 - <http://wiki.ubuntuusers.de/Java/Installation>
 - <https://wiki.archlinux.de/title/Java>

JDK unter macOS

- Java SE Development Kit 8
 - <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

- Das JDK 8 benötigt mindestens OS X 10.8 oder höher
 - https://docs.oracle.com/javase/8/docs/technotes/guides/install/mac_jdk.html

- Mit Homebrew:
 - `brew cask install java`

Kommandozeile

■ Eingabebereich im Textmodus für die Steuerung des Betriebssystems

■ Windows

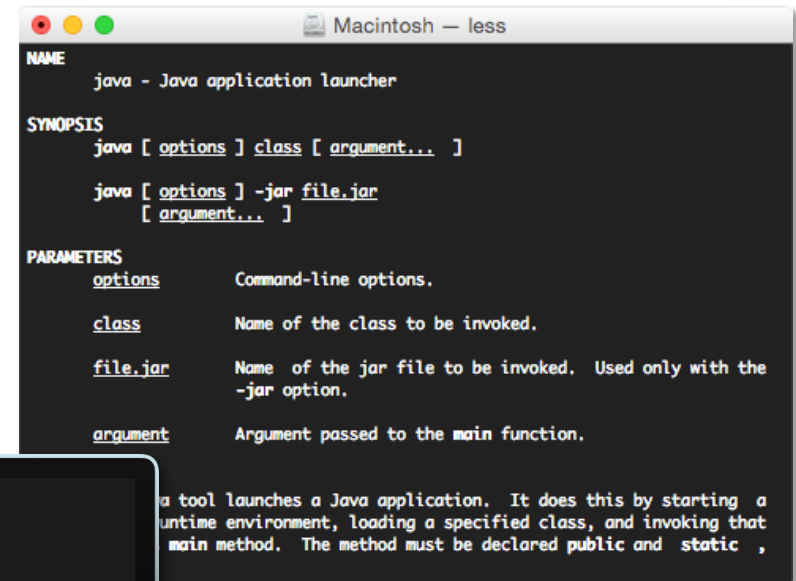
- Win + R, „cmd“, Enter

■ Ubuntu

- Strg + Alt + T

■ macOS

- cmd + Leertaste, „Terminal“, Enter



```

Macintosh — less
NAME
    java - Java application launcher

SYNOPSIS
    java [ options ] class [ argument... ]

    java [ options ] -jar file.jar
    [ argument... ]

PARAMETERS
    options      Command-line options.
    class        Name of the class to be invoked.
    file.jar     Name of the jar file to be invoked. Used only with the
                 -jar option.
    argument     Argument passed to the main function.

a tool launches a Java application. It does this by starting a
runtime environment, loading a specified class, and invoking that
main method. The method must be declared public and static ,

```



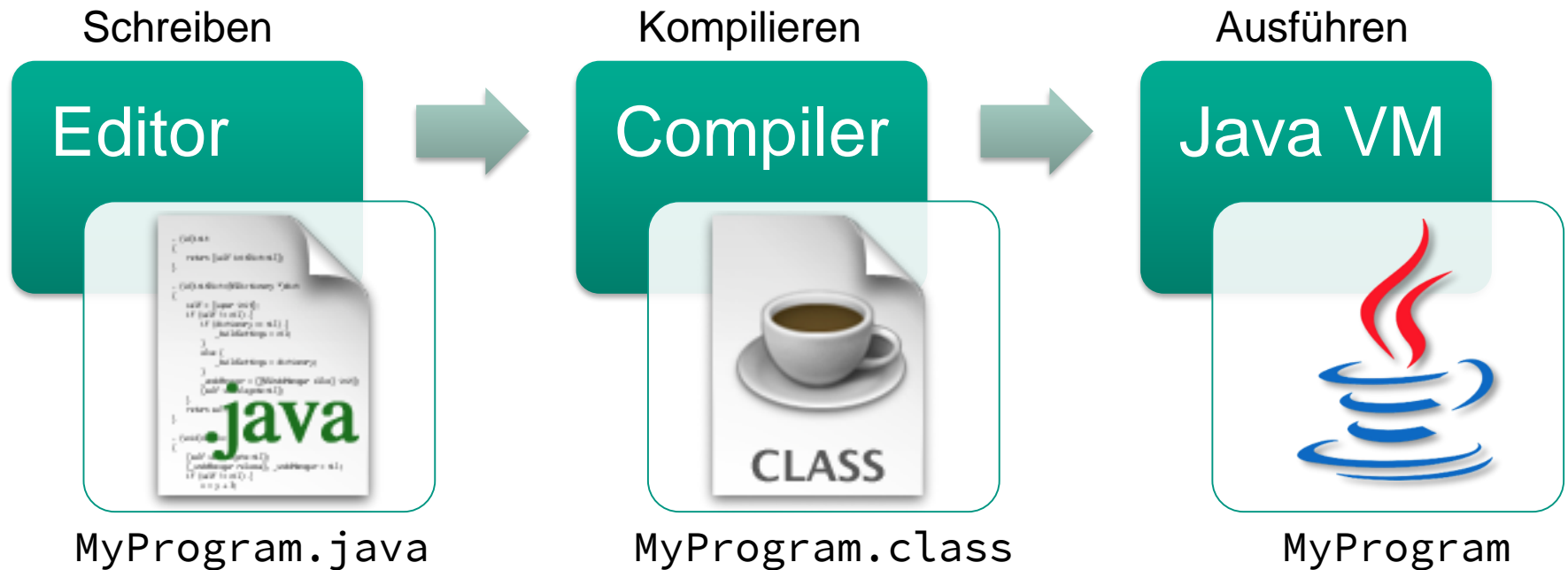
Kleines Kommandozeilen Einmaleins

| Befehl | Beschreibung |
|--|---|
| <code>ls</code> | Zeigt den Inhalt des Verzeichnisses an in dem man sich gerade befindet (Linux/Unix) |
| <code>dir</code> | Zeigt den Inhalt des Verzeichnisses an in dem man sich gerade befindet (Windows) |
| <code>cd <Unterverzeichnis></code> | Wechselt in ein Verzeichnis tiefer |
| <code>cd ..</code> | Wechselt in ein höheres Verzeichnis |
| <code>cd ~</code> | Wechselt in das eigene Benutzerverzeichnis |

■ Weiter Informationen auf:

- <https://win10faq.com/command-prompt-windows-console/>
- <http://wiki.ubuntuusers.de/Terminal>
- <http://www.apfelwiki.de/Main/Terminal>

Java Arbeitsablauf



Schreiben

■ Schreiben des Java Quelltexts mit einem einfachen Texteditor

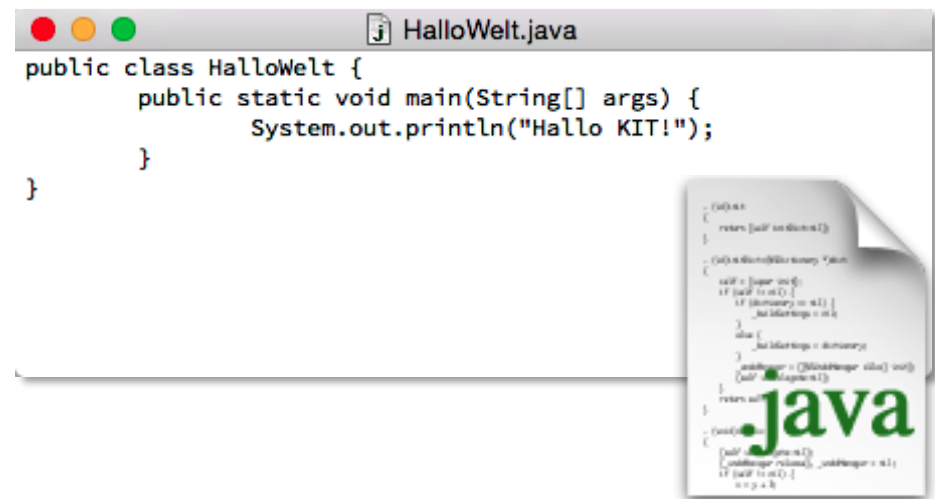
- Windows
 - Notepad, Notepad++, ...
- Linux
 - gedit, vim, nano, ...
- macOS
 - TextEdit, TextMate, ...

■ Kein Word, Writer, Pages, ...

■ Dateiname = Klassenname.java

■ Vorerst keine IDEs

- ~~Eclipse~~
- ~~IntelliJ IDEA~~



Programmelemente

■ Anweisungen

- Elementarste ausführbare Programmelemente
- Werden mit Semikolon beendet

■ Blöcke

- Reihe von Anweisungen
- Wirkt nach außen hin wie eine Anweisung
- Werden von geschwungenen Klammern umgrenzt

■ Methoden

- Spezieller Block, welcher über seine Signatur aufrufbar ist

■ Klassendefinitionen

- Grundlegende Strukturierungseinheit eines objektorientierten Programms

Das erste Programm

```
public class HalloWelt {  
    public static void main(String[] args) {  
        // Ich bin ein Kommentar  
        System.out.println("Hallo Welt!");  
    }  
}
```

Diagram labels and arrows:

- Klasse**: points to `public class`
- Methode**: points to `main`
- Kommentar**: points to `// Ich bin ein Kommentar`
- Anweisung**: points to `System.out.println("Hallo Welt!");`
- Block**: points to the curly braces of the `main` method

main-Methode

```
public class Untitled {  
    // ...  
    public static void main(String[] args) {  
        // Die auszuführenden Anweisungen  
    }  
    // ...  
}
```

- Jedes **ausführbare** Programm muss die main-Methode enthalten
- Sie ist der Einstiegspunkt in das Programm
- Es darf immer nur eine main-Methode je Programm geben
- Das Programm beendet sich beim Verlassen der main-Methode

Kompilieren

- Der Java Compiler prüft auf syntaktische Korrektheit des Quelltexts und übersetzt den Quelltext (.java) in einen ausführbaren Bytecode (.class)
 - `Klassenname.java` \Rightarrow `Klassenname.class`
- Aufruf in der Eingabeaufforderung (Groß-/Kleinschreibung beachten):
 - **`javac Klassenname.java`**
- Bei erfolgreichem Übersetzen wird keine Nachricht angezeigt
- Bei syntaktischen Fehlern wird ein Fehlerreport angezeigt
- Weitere Informationen mit:
 - `javac -help`
 - `man javac`

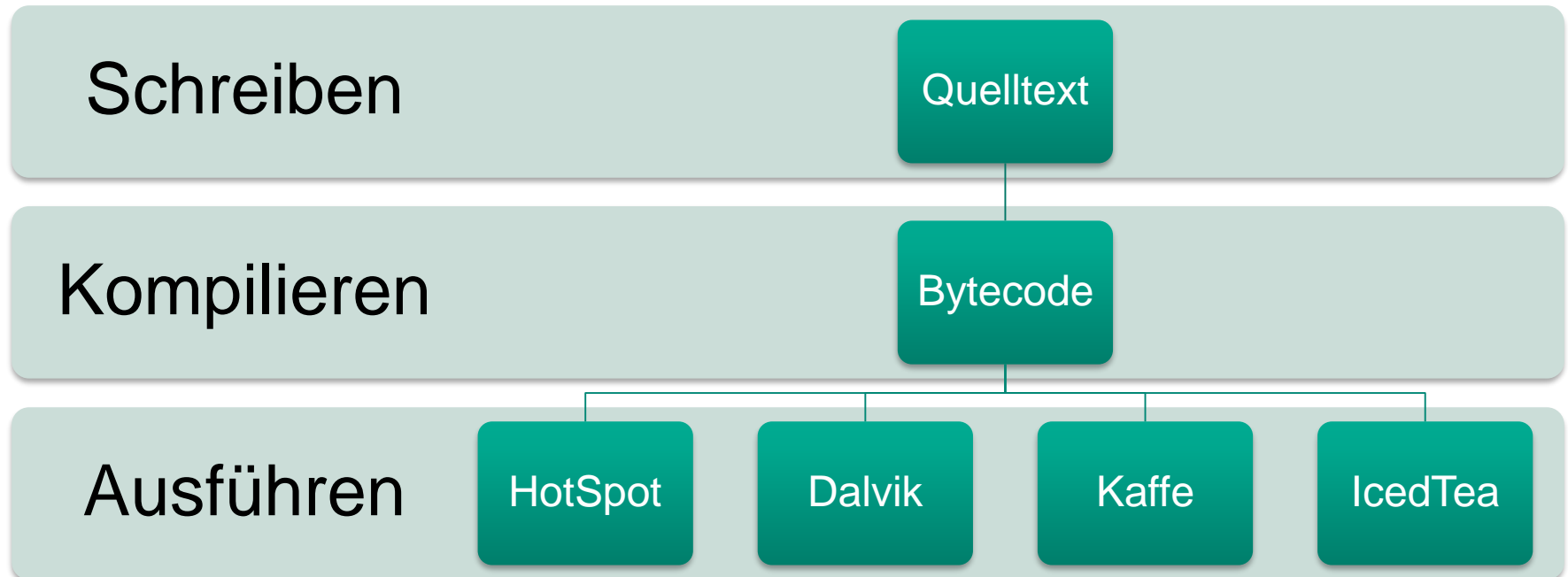


Java Bytecode

- Zwischensprache vom Java-Compiler generiert
- Befehlssatz für die Java Virtuelle Maschine (JVM)
- Erst die JVM erzeugt aus dem Bytecode den eigentlichen Maschinencode
- Unabhängig von der zugrundeliegenden Hardware
- Anzeigen des Bytecode mit:
 - **javap -c Klassenname**

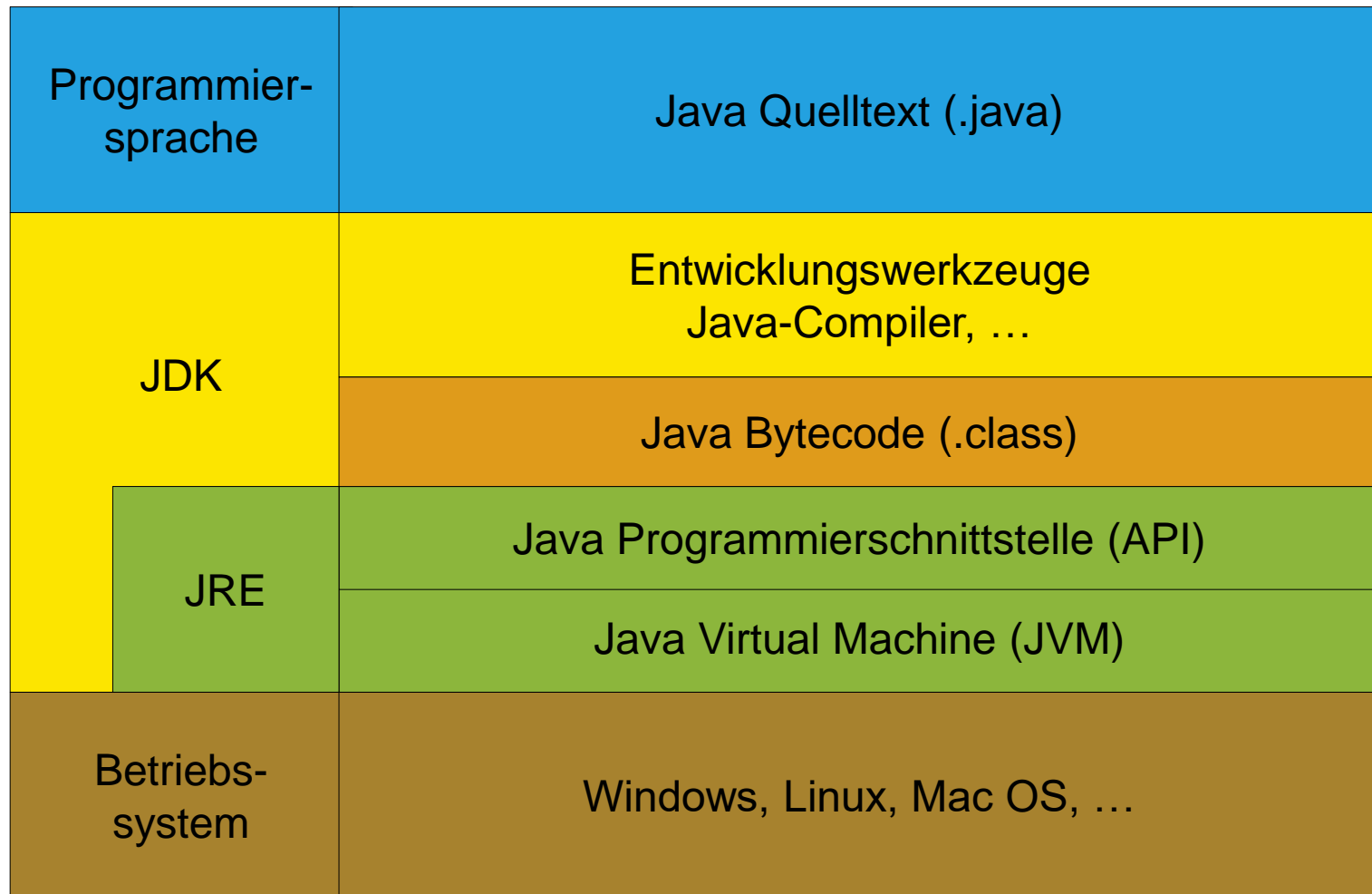
Ausführen

- Ausführen des Bytecodes (.class) in der Java Virtuellen Maschine (JVM) mit dem Befehl:
 - **java Klassenname**
- Achtung, die Groß-/Kleinschreibung beachten und nicht mit der Dateiendung aufrufen:
 - ~~java Klassenname.class~~
- Funktioniert nur mit Klassen, welche die sogenannte main-Methode enthalten
- Weitere Informationen wieder mit:
 - `java -help`
 - `man java`

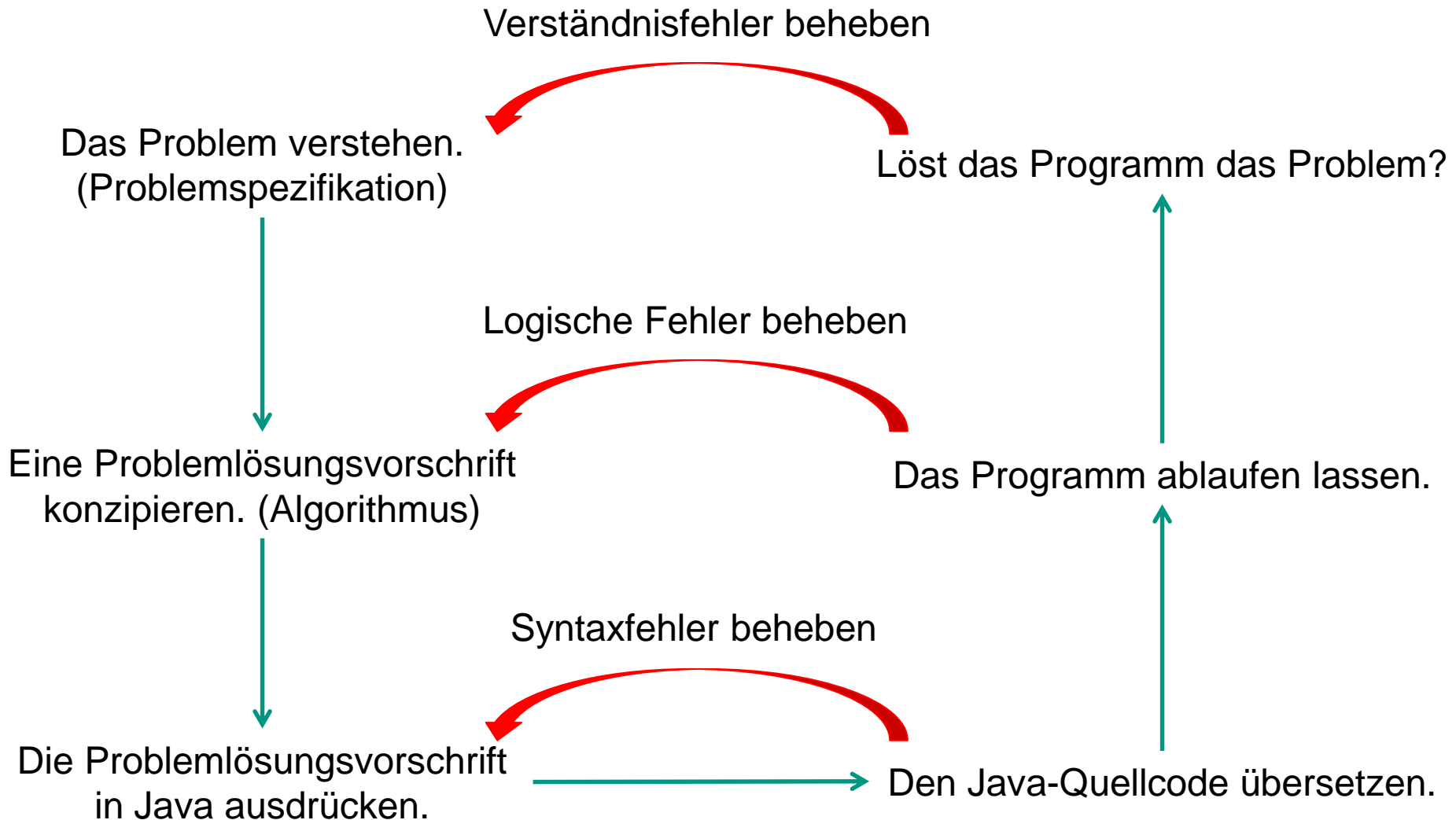


- „Write once, run anywhere“
- Verantwortlich für die Ausführung des Bytecodes
- Dient als Schnittstelle zur Hardware und zum Betriebssystem
- Meist in C und C++ geschrieben

Aufbau der Java-Technologie



Der Weg zu einem (Java-)Programm



Nächste Woche

- Datentypen und Operatoren
- Klassen und Objekte

Fragen?

Vielen Danke für eure Aufmerksamkeit!

