

PM2 – Programmieren in Java

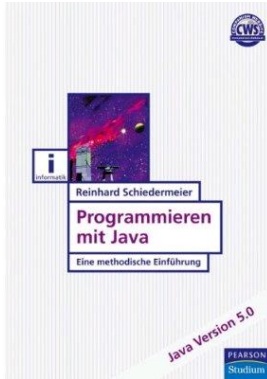
V0 Überblick über den Semesterstoff

Semesterfahrplan

- V0-Überblick und Literatur
- V1-Unterschiede Ruby Java am Beispiel
- V2-Basisdatentypen und Operatoren
- V3- Kontrollstrukturen
- V4-Arrays
- V5-Zeichenketten und reguläre Ausdrücke
- V6-Packages und Sichtbarkeiten
- V7- JUnit
- V8-Klassen-Typen-Interfaces
- V9- Factory Pattern
- V10-Collections – Verwenden von Lambdas und Streams
- V11-Aufzählungstypen
- V12-Generics
- V13-Innere Klassen
- V14-Vertiefung Lambdas und Streams (evtl.)
- V15-Fehlerbehandlung
- V16-GUI-Entwicklung mit JavaFX / Observer Pattern
- V17-IO (wenn noch Zeit ist)
- V18-Design Patterns (wenn noch Zeit ist)
- V19- Prüfungsvorbereitung

Die Beziehung zwischen Vorlesungsfolien und Vorlesungstermin ist nicht zwingend 1:1!

Literatur - Lehrbücher



Programmieren mit Java I und II
Reinhard Schiedermeier (bis Java 7)



Einführung in die Programmierung mit Java
Robert Sedgewick / Kevin Wayne (kein Java 8)



<http://openbook.galileocomputing.de/javainsel8/>

Referenzen – Sprachspezifikation und API Dokumentation

The Java® Language Specification Java SE 8 Edition

James Gosling
Bill Joy
Guy Steele
Gilad Bracha
Alex Buckley

2015-02-13

<http://docs.oracle.com/javase/specs/index.html>

Java™ Platform, Standard Edition 8 API Specification

This document is the API specification for the Java™ Platform, Standard Edition.

See: Description

Profiles

- compact1
- compact2
- compact3

Packages

Package	Description
java.applet	Provides the classes for an applet and the classes to communicate with the host environment.
java.awt	Contains all of the classes for user interfaces and graphics and image processing.

<https://docs.oracle.com/javase/8/docs/api/>

Online Tutorials

- Oracle Tutorial: <https://docs.oracle.com/javase/tutorial/>
- <http://coreservlets.com/> (Marty Hall)

Etwas Geschichte: Oak als Laufzeitumgebung für Konsumelektronik

1990	Bill Joy beschreibt die Grundzüge einer neuen objektorientierten Sprache basierend auf C++
zeitgleich	James Gosling entwickelt Oak nach Erfahrung mit der Entwicklung eines SGML Interpreters in C++ (Fensteranekdote)
Dezember 1990	Patrick Naughton startet das Green Projekt mit James Goslin und Mike Sheridan. Duke verdanken wir diesem Projekt.
Projekt Green	Ziel: Software für interaktives Fernsehen und Geräte der Konsumelektronik. Technik: BS Green-OS, Oak, grafische Benutzeroberfläche, Hardwarekomponenten. Gosling: Originalcompiler in C; Naughton, Gosling und Sheridan: Runtime-Interpreter in C.
1992	*7 (Star Seven) – portabler Mikrocomputer zur Steuerung von Haushaltsgeräten in Oak geschrieben. Interne nicht marktreife Studie.
November 1992	Scott McNealy (Sun Chef) Gründung von First Person, Inc. Vermarktung *7
1993	Time Warner Anfragen bezüglich eines Systems für Set-Top-Boxen.

Etwas Geschichte: Oak wird zu Java, einer Sprache für das Web

ab 1993	Konzentration auf Entwicklung für das Web. Laden von Programmcode über das Netzwerk. Sandbox Modell: Fehlerhafte und bösartige Programme dürfen keinen Schaden anrichten → C++ scheidet als Sprache aus.
	Aus Oak wird Java (Copyright Gründe) im Café „Java City – roasters of fine coffee“ im Menlo Park
SunWorld 95	Patrick Naughton's "WebRunner" (in "HotJava" umbenannt) wird vorgestellt
1995	Netscape lizenziert die Java-Technologie. Netscape Navigators 2.0 enthält erste Implementierung.
1996	Freigabe des JDK 1.0 enthält Java-Applikationen und Web-Applets.
1996	Gründung der Firma JavaSoft.
ab 1996	Entwicklung einer Reihe interaktiver Programme für das Web auf Basis von Applets. Applets werden in HTML eingebettet. Der Browser lädt den Sourcecode über das Netz. Applets haben eingeschränkten Zugriff auf das System, auf dem sie ausgeführt werden. Z.B. keinen Zugriff auf das Dateisystem. Microsoft unterstützt Applets nur bis zur Version 1.1.4

Etwas Geschichte: Java als Sprache für Serveranwendungen.

heute

Java Sprache für die Serverentwicklung: Plattformen für große Unternehmensanwendungen in Java (Enterprise Service Applications / SOA).

Argumente für Java:

- Write once, run anywhere
- robust durch eigene Speicherverwaltung

Java Sprache für serverseitige Webapplikationen. (Webapplication Server: Tomcat, Frameworks: Struts, MyFaces).

Java hat sich für klassische Client-Anwendungen auf dem breiten Markt nicht durchgesetzt. Argumente:

1. Java ist zu langsam
2. GUIs in Java unterschieden sich anfänglich zu sehr vom nativen "Look and Feel"

zu 1.) JIT-Compilern und Hotspot-Technologie haben die Geschwindigkeit deutlich verbessert. Eclipse ist z.B. in Java geschrieben.

Java Versionen

Version	Jahr	Eigenschaften und wesentliche Erweiterungen
1.0	1996	Standardpakete java.lang, java.io, java.util, java.net, java.awt, java.applet
1.1	1997	Ereignisbehandlung (Callback und Listener), I18N, RMI, Beans, jar, JDBC, Serialisierung, Sicherheitsfunktion (Signieren von jars und Applets)
1.2	1998	Just In Time (JIT) Compiler, Erweiterung der Collection Klassen, Swing, CORBA Integration, Java2D
1.3	2000	Hotspot Optimierung des JIT
1.4	2002	Verbesserung der serverseitigen APIs (z.B. Servlets, JSPs, J2EE), Assertions
5	2004	Generics, Annotationen, Autoboxing (-unboxing), Enumerations
6	2006	WEB 2.0: Frameworks zur Integration von Python, PHP, JavaScript, Ruby
7	2010	Strings in switch-case; try-with-resources – verbessertes Exception-Handling; Unterstützung für dynamisch typisierte Sprachen im Compiler.
8	2014	Lambdas & Streams
9	2017	Modulsystem Jigsaw
10	2018	Typinferenz für lokale Variablen
11	2018	angekündigt für September

Entwicklungsumgebung

- IntelliJ Ultimate (Java): Lizenz frei erhältlich für Studierende
- **ACHTUNG: UNBEDINGT MIT UTF-8 arbeiten!**
(<https://www.jetbrains.com/help/idea/configuring-individual-file-encoding.html#options>)
- GUI-JavaFX:
 - SceneBuilder für die korrekte JavaVersion herunterladen und installieren: <https://gluonhq.com/products/scene-builder/>
 - IDE Plugin für IntelliJ installieren (aus IntelliJ heraus): siehe <https://gluonhq.com/get-started/ide-plugins/>
 - IDE neu starten, das wars.
 - Evtl. "Live Templates" nachinstalliert (<https://gluonhq.com/javafx-live-templates-for-intellij-idea/>). Damit wird das Entwickeln der Controller etwa komfortabler
- Aufbau der Quelltexte:
 - 1 IntelliJ Projekt mit Modulen für die einzelnen Vorlesungen
 - Commons-Module unbedingt mit importieren, das einige Module Abhängigkeiten zu dem Modul haben.