

Praktikum "SEP: Java-Programmierung"

WS 2018/19

Organisatorisches

Thomas Lemberger und Martin Spießl
Basierend auf Folien von Matthias Dangl und Karlheinz Friedberger

- ▶ Anmeldung zur Veranstaltung im UniWorX (erledigt via Zentralanmeldung)
- ▶ Wichtige Informationen auf Vorlesungsseite (im UniWorX verlinkt)
- ▶ Unbedingt regelmäßig Vorlesungsseite besuchen!

- ▶ Mail-Liste für weitere Lehrstuhl-Infos

- ▶ Vorlesung, Aufgabenstellung und Besprechung:
 - ▶ Montag 12:15 – 13:45 Uhr
 - ▶ Amalienstr. 73A, Raum 112
- ▶ Personen
 - ▶ Thomas Lemberger
 - ▶ Martin Spießl

Erste Aufgabe für euch:
Baseline für Antworten im Plenum

- ▶ gängige Algorithmen und Datenstrukturen
- ▶ Objekt-orientiertes Programmieren
(Vererbung, Information Hiding,...)
- ▶ Design-Pattern
- ▶ Nebenläufige Programmierung
- ▶ Testen und Debuggen

- ▶ Umsetzung in Java

Empfohlene Voraussetzungen I

- ▶ Fundierte Erfahrung mit Java
 - ▶ Udacity Course: Java Programming Basics
<https://tinyurl.com/ud4c1tyJava>
 - ▶ Buch: Christian Ullendbloom. Java ist auch eine Insel
<http://openbook.rheinwerk-verlag.de/javainsel/>

- ▶ Kenntnisse von VCS, bei uns: Git(Lab)
 - ▶ Simple words for a GitLab Newbie:
<https://tinyurl.com/g1t14bNewbie>
 - ▶ Udacity Course: Version Control with Git
<https://tinyurl.com/ud4c1tyGit>
 - ▶ Buch: Scott Chacon, Ben Straub. Pro Git
<https://tinyurl.com/pr0G1t>

Empfohlene Voraussetzungen II

- ▶ Nutzung der Kommandozeile
 - ▶ Udacity Course: Shell Workshop
<https://tinyurl.com/ud4c1tyShell>
 - ▶ Buch: William Shotts. The Linux Command Line
<https://tinyurl.com/l1nuxLC3>

- ▶ Design-Patterns und Code-Smells
 - ▶ Buch: Gang of Four. Design Patterns: Elements of Reusable Object-Oriented Software.
 - ▶ Website: z. B. Source Making
<https://sourcemaking.com/>

- ▶ Theoretische Grundlagen der Programmierung und von Algorithmen und Datenstrukturen
- ▶ Entsprechende Vorlesungen, siehe Veranstaltungswebseite für detaillierte Liste

- ▶ Jedes Mal: Aufgabenbesprechung
- ▶ Kurze Einführungen:
 - ▶ Codestyle
 - ▶ Module/Kapselung/Objektorientiertes Arbeiten in Java
 - ▶ Git(Lab)?
 - ▶ Evtl. Nutzung von Analyse-Tools/Gitlab CI
 - ▶ Wünsche?

- ▶ Anhand von 4 Programmieraufgaben
 - ▶ 3 Einzelabgaben
 - ▶ 1 Teamabgabe (5–7 Personen)
- ▶ Codebewertung nach folgenden Kriterien:
 - ▶ Funktionalität
 - ▶ Struktur
 - ▶ Layout (Kommentare u. Formatierung)
- ▶ Zusätzlich bei letzter Aufgabe: Bewertung der Teamarbeit
- ▶ jeweils **eine** Deadline, keine Nachholtermine

- ▶ Disclaimer ausfüllen!

- ▶ Aufgaben von uns
- ▶ Lösungen von euch
- ▶ Abgabe über den Praktomat
- ▶ Fragen und Besprechung in der „Vorlesung“

- ▶ <https://praktomat.sosy.ifi.lmu.de>
- ▶ Registrierung mit LMU-Email-Adresse (name@campus.lmu.de)
 - ▶ Anmeldung ab morgen möglich
 - ▶ Deadline: 23.10. (nächster Dienstag)
- ▶ Aufgabenstellung und Abgabe im Praktomat
- ▶ Automatische Tests bei Abgabe:
 - Compiler, CheckStyle,
 - Plagiatstest, öffentliche und geheime Tests, ...
- ▶ Mehrfache Abgaben vor Deadline möglich, nur letzte Abgabe wird bewertet

	Aufgabe	Zeitraum	#Wochen
1.	Trie	16.10. – 05.11.	3
2.	Merkle Tree	12.11. – 03.12.	3
3.	TBD	03.12. – 17.12.	2
4.	TBD	17.12. – 04.02.	5

- ▶ Aufgabenstellungen werden zum jeweiligen Starttermin veröffentlicht.
- ▶ Abgaben vor jeweiliger Deadline sind möglich!
- ▶ Aufgaben 1–3 in Einzelarbeit.
- ▶ Aufgabe 4 in Teamarbeit.