

Softwaredesign

Julian Huber & Matthias Panny

Allgemeine Info

- Matthias Panny
(Matthias.Panny@mci.edu)
- BSc, MSc in Mechatronik, MCI
- Wissenschaftlicher Mitarbeiter im Studiengang **Mechatronik**



- Julian Huber (Julian.Huber@mci.edu)
- Dr.rer.pol in Information Systems, KIT
- BSc, MSc in Wirtschaftsingenieurwesen, TU Berlin, KIT
- Hochschullektor im Studiengang **Smart Building Technologies**



Ausgangslage

- Kenntnisse in C
- Kenntnisse in C++
 - Objektorientierte Programmierung
 - Templates
- Spiel als erstes größeres Projekt umgesetzt

Ziel

- Grundlagen in **Python**
- Versionskontrolle mit **git** & kollaboratives Arbeiten
- **Algorithmen**, deren Analyse & Anwendung
- Wie entwickelt man auf moderne/sinnvolle Art Software?
- Anwendung des Gelernten in einer **Case Study**
- →  **keine reine Programmier-LV**

Prüfungsmodalität

- Abschlussprojekt als Gruppenarbeit: 100%
- Hausübungen
 - 15 Hausübungen gesamt
 - 10% Abzug pro fehlender Hausübung ab 2 fehlenden
- Gesamt sind 60% notwendig für eine positive Note
- Nachklausur: mündliche Prüfung

Zeitaufwand

- Lehrveranstaltung mit **4 SWS & 5 ECTS**
- 4 SWS = 45h gemeinsame Vorlesungszeit
- 5 ECTS = 125h Zeitaufwand (im Durchschnitt)
- $125h - 45h = 80h$ **Zeitaufwand für Hausübungen & Projekt**

- Zu den meisten Einheiten sind Hausübungen zu absolvieren → letzte Folie mit  markiert
- Die Angabe erfolgt mittels **Jupyter Notebook**
- Die Abgabe erfolgt als **ausgeführtes** Notebook über das **Sakai-Assignment** → ordnungsgemäß und fristgerecht

Benotung der Hausübungen

- **Mehr als im trivialen Ausmaß** mit der HÜ beschäftigt → **0%**
- Nicht mit der Hausübung beschäftigt → **-10%**
- Eine vergessene/nicht gemachte HÜ als Toleranz

- 🧐: Folien mit Inhalten die über das Minimum hinausgehen → je nach Zeit für Interessierte im Selbststudium
- 🔄: Wiederholung von Inhalten aus dieser LV oder anderen LVs
- 🖋️: Aufgaben in der LV → machen wir teils gemeinsam
- Foliensätze mit **xx_Ax_Name_des_Foliensatzes** → Appendix
- 🏆: Kurze Angabe zur Hausübung

- Diese LV findet in der aktuellen Form das zweite Mal statt
- Bitte gebt uns auch schon **während der LV** Feedback - am Ende des Semesters ist es zu spät um für euch noch viel zu ändern

Literaturempfehlungen

- Besonders empfohlene Literatur ist rot hervorgehoben

Literatur

- C. Althoff, **The Self-Taught Computer Scientist**, 2021.
- A. Shvets, **Dive Into Design Patterns**, 2019.
- E. Matthes, **Python crash course: a hands-on, project-based introduction to programming**, 2016.
- D. E. Knuth, **The Art of Computer Programming: Volume 1-4A/B: 1997-2022**.
- B. Miller & D. Ranum, **Problem Solving with Algorithms and Data Structures using Python**, 2011.
- J. VanderPlas, **Python Data Science Handbook**, 2016.
- H.-P. Halvorsen, **Python for Science and Engineering**, 2020.
- N. Wirth, **Algorithms and Data Structures**, 1985.
- N. Wirth, **Algorithms + Data Structures = Programs**, 1976.
- K. Passig, J. Jander. **Weniger schlecht Programmieren**, 2013.