

Einführung in die Programmierung – Organisation

Manfred Hauswirth | Open Distributed Systems | Einführung in die Programmierung, WS 25/26

Rückblick

- VL 0 „Organisation und Inhalt“: Ablauf der Vorlesung, Termine
- VL 1 „Algorithmen, Pseudocode, Sortieren I“: Insertion Sort
- VL 2 „Algorithmen, Pseudocode, Sortieren II“: Selection Sort, Bubble Sort, Count Sort
- VL 3 „Laufzeit und Speicherplatz“: Laufzeitanalyse der vorgestellten Sortiervverfahren
- VL 4 „Einfache Datenstrukturen“: Arrays, verkettete Listen, Structs in C, Stack, Queue
- VL 5 „Bäume“: Binärbäume, Baumtraversierung, Laufzeitanalyse Baumoperationen
- VL 6 „Teile und Herrsche I“: Einführung der algorithmischen Methode, Merge Sort
- VL 7 „Korrektheitsbeweise“: Rechnermodel, Beispielbeweise
- VL 8 „Dateien in C“: Dateien, Dateisysteme, Verzeichnisse, Dateiverwaltung mit C
- VL 9 „Prioritätenschlangen/Halden/Heaps“: Heap Sort, Binärer Heap, Heap Operationen
- VL 10 „Fortgeschrittene Sortiervverfahren“: Quick Sort, Radix Sort
- VL 11 „AVL Bäume“: Definition, Baumoperationen, Traversierung
- VL 12 „Teile und Herrsche II“: Generalisierung des algorithmischen Prinzips, Mastertheorem
- VL 13 „Q & A“: Offene Vorlesung/Wiederholung

Heutige Themen

- Weitere Organisation
- 1. Vorlesung
 - Motivation: Datenstrukturen und Algorithmen
 - Algorithmen und Pseudo-Code
 - Unser erster Sortieralgorithmus

Weitere Organisation

- Kenntnisse
 - elementarer Datenstrukturen
 - elementarer Such- und Sortiervverfahren
- Fähigkeiten
 - Probleme und Strukturen (wieder) zu erkennen
 - für ein gegebenes Anwendungsproblem die geeignete Datenstruktur zu wählen
 - den Aufwand (Komplexität) eines Algorithmus bzw. eines C-Programms abzuschätzen

Lernziele

- Datenstrukturen
 - Listen
 - Queue, Stack und Heap
 - Bäume
- Elementare Algorithmen
 - Suchen
 - Sortieren
- Algorithmen
 - Aufwandsabschätzung
 - Korrektheit

Lehr- und Lernkonzept

- Vorlesung
 - Vorstellung der Konzepte & Beispiel-Programme
- Tutorien
 - (Vor-)Besprechung der Hausaufgaben & Codebeispiele
- Betreute Rechnerzeiten:
 - Hilfestellung beim Programmieren, inkl. Fehlersuche, online und in Präsenz (siehe ISIS)
- Hausaufgaben:
 - eigenständige Auseinandersetzung mit den Konzepten, als Einzelabgaben (wie auf dem jeweiligen Blatt angegeben)

Ablauf im Detail

- **Veranstaltung besteht aus:**

- 1. Programmierkurs**

- Vorstellung der Konzepte
- **Blockveranstaltung (täglich), 13.10. – 24.10.2025, 12:15 – 13:45 Uhr**
- Folgende Vorlesungen finden in diesen zwei Wochen nicht statt:
 - Rechnerorganisation
 - Informatik Propädeutikum

- 2. Einführung in die Programmierung (IntroProg):**

- Grundlegende Datenstrukturen
- Algorithmen – am Beispiel von Listen, Bäumen, und Sortieren
- **Dauer: Rest des Semesters**

Termine IntroProg

- Ab dem **27.10.2025** regulärer Vorlesungsbetrieb
 - Vorlesung, Tutorien und Rechnerübungen (wöchentlich)
- Vorlesung
 - Donnerstags, jeweils 14:15 – 15:45 Uhr, **H0105**
 - **Am 06.11. und 20.11. findet die VL nur in Zoom statt.**
 - Zoom: <https://s.fhg.de/2025-VL-Introprog>
- Tutorien:
 - Beginn: **27.10.2025** (**Termine in ISIS**)
- Rechnerübung:
 - Beginn **27.10.2025** (**Termine in ISIS**)

Lehr- und Lernkonzept

Leistungen der Portfolioprüfung

- Hausaufgaben im Programmierkurs (Programmierung)
 - eigenständige Auseinandersetzung mit den Konzepten
 - **15% der Gesamtnote**
- Hausaufgaben während des Semesters (Theorie)
 - eigenständige Auseinandersetzung mit den Konzepten
 - **20% der Gesamtnote**
- praktischer Test vor Semesterende (60min)
 - **30% der Gesamtnote**
- schriftlicher Test am Semesterende (60min)
 - **35% der Gesamtnote**

Anmeldefristen und Tests

- ISIS für Vorlesungsmaterial – **am besten sofort**
- Modulanmeldefrist via Moses oder Prüfungsamt (Portfolioprüfung)
Anmeldung ab 14.10.2025, 00:00, und bis 09.11.2025, 23:59
Abmeldung (Rücktritt) bis zum 13.11.2025, 23:59
- Funktionstest für Praxistest – auf Laptops, HE 101
 - **Fr. 12.12.25, 14:00 - 16:00 (freiwillig, aber empfohlen)**
- Praxistest – auf Laptops, HE 101
 - **1. Termin (empfohlen): Sa. 17.01.26, 09:00 - 18:00**
 - **2. Termin (Wiederholungsmöglichkeit *): Fr. 27.03.26, 10:00 - 19:00**
- Theorietest – mit Blatt und Stift, auch an der Uni
 - **1. Termin (empfohlen): Mo. 23.02.26, 15:00 – 18:00**
 - **2. Termin (Wiederholungsmöglichkeit *): Mi. 25.03.26, 12:30 - 17:30**
- **Beachten Sie die Ankündigungen in ISIS !**

Anmeldefristen und Tests

- **Beachten Sie die Ankündigungen in ISIS !**

* Sie dürfen jeweils nur an einem Praxis- und Theorietest teilnehmen.

Sie dürfen sich nur zum zweiten Termin anmelden, falls Sie insgesamt < 50 Punkte für alle Leistungen erreicht haben.

Dann dürfen Sie wählen, welche Tests Sie wiederholen möchten.

Die Semesterpunkte werden angerechnet.

Abgabe der Programmieraufgaben

- Ablauf zum Hochladen: siehe ISIS und Blatt 10 Programmierkurs
- Freischaltung Aufgaben: **Donnerstag, 16:00 Uhr (nach VL)**
- Deadline für die Abgabe: **Freitag drei Wochen nach der Ausgabe des ersten Blattes der Kategorie, 20:00 Uhr**
- **Programmierkurs (15%)**
 - Blätter 1-10 des Programmierkurses. Ausgabe: 13.10.-24.10.2025, letzte Abgabe: 14.11.2025
- **Theoriaufgaben (20%)**
 - Block A – Ausgabe: 30.10.2025, Abgabe: 21.11.2025
 - Block B – Ausgabe: 20.11.2025, Abgabe: 12.12.2025
 - Block C – Ausgabe: 11.12.2025, Abgabe: 23.01.2026
 - Block D – Ausgabe: 15.01.2026, Abgabe: 06.02.2026
- **Rest:**
 - Praxis-Test (30%)
 - Theorie-Test (35%)

Einzelabgabe – wichtige Hinweise

- Einzelabgabe
 - Jede/r Studierende erarbeitet eine eigene Lösung und gibt diese ab!
 - Diskussionen von Lösungswegen, Herangehensweisen und Hilfestellungen sind erlaubt und sogar erwünscht!
 - Aber **Weitergabe von Lösungsteilen ist keine Hilfestellung**, da dies nicht dazu führt, ein **eigenes Verständnis zu entwickeln!**
- Regeln
 - Zwei identische Abgaben
 - ⇒ Eine ist ein **Plagiat!**
 - ⇒ Das ist ein **Täuschungsversuch.**
 - ⇒ **Beide Abgaben gelten als nicht bearbeitet**, da generell der/die Originalautor/in nicht ermittelbar ist.
 - **Wiederholungsfall ⇒ Nichtbestehen – wegen Täuschung**

Identische Abgabeteile

- Abgaben werden als identisch betrachtet, wenn sie sich, u.a. nur in den
 - Variablennamen
 - Kommentaren
 - Einrückungenunterscheiden.

**Hinweis: Wir benutzen Plagiatcheckertools!
Zusammen mit manueller Überprüfung**

Literatur - Beispiele

- **C**
 - Kernighan, Programmieren in C, 1990
- **Algorithmen und Datenstrukturen**
 - Cormen, T.H.; Leiserson, C.E.; Rivest, R.L.; Stein, C.: Introduction to Algorithms, 3. Aufl. MIT Press Cambridge, 2009
 - Sedgewick, R.: Algorithms in C, Addison-Wesley, 2005
 - Goodrich, M. Tamassia, R.: Data Structures and Algorithms in C++, John Wiley
- **Systemsoftware**
 - Randal E. Bryant, David R. O'Hallaron „Computer Systems: A Programmer's Perspective“, Prentice Hall

- Modern C, J. Gustedt
 - <https://gustedt.gitlabpages.inria.fr/modern-c/>
- Beej's Guide to C Programming, Brian “Beej” Hall
 - <http://beej.us/guide/bgc/>

Ausblick

VL 0 „Organisation und Inhalt“: Ablauf der Vorlesung, Termine

VL 1 „Algorithmen, Pseudocode, Sortieren I“: Insertion Sort

VL 2 „Algorithmen, Pseudocode, Sortieren II“: Selection Sort, Bubble Sort, Count Sort

VL 3 „Laufzeit und Speicherplatz“: Laufzeitanalyse der vorgestellten Sortiervverfahren

VL 4 „Einfache Datenstrukturen“: Arrays, verkettete Listen, Structs in C, Stack, Queue

VL 5 „Bäume“: Binärbäume, Baumtraversierung, Laufzeitanalyse Baumoperationen

VL 6 „Teile und Herrsche I“: Einführung der algorithmischen Methode, Merge Sort

VL 7 „Korrektheitsbeweise“: Rechnermodel, Beispielbeweise

VL 8 „Dateien in C“: Dateien, Dateisysteme, Verzeichnisse, Dateiverwaltung mit C

VL 9 „Prioritätenslangen/Halden/Heaps“: Heap Sort, Binärer Heap, Heap Operationen

VL 10 „Fortgeschrittene Sortiervverfahren“: Quick Sort, Radix Sort

VL 11 „AVL Bäume“: Definition, Baumoperationen, Traversierung

VL 12 „Teile und Herrsche II“: Generalisierung des algorithmischen Prinzips, Mastertheorem

VL 13 „Q & A“: Offene Vorlesung/Wiederholung