

Einführung in die Programmierung – Organisation

Manfred Hauswirth | Open Distributed Systems | Einführung in die Programmierung, WS 23/24

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 Sortierverfahren
- 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
- Tutorium-Videos
 - Kleine Videosequenzen zur Erklärung des jeweiligen Themas der Hausaufgaben
- Tutorien
 - (Vor-)Besprechung der Hausaufgaben & Codebeispiele
- Großübungen
 - Nachbesprechung der Hausaufgaben
- 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), 16.10. – 28.10.2023, 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 **30.10.2023** regulärer Vorlesungsbetrieb
 - Vorlesung, Tutorien und Rechnerübungen (wöchentlich)
- Vorlesung
 - Donnerstags, jeweils 14:15 – 15:45 Uhr, H0105
 - Zoom: <https://s.fhg.de/2023-VL-Introprog>
- Tutorien:
 - Beginn: **30.10.2023** ([Termine in ISIS](#))
 - Zusätzlich Videos auf ISIS verfügbar
- Rechnerübung:
 - Beginn **30.10.2023** ([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 (Programmierung und Theorie)
 - eigenständige Auseinandersetzung mit den Konzepten
 - **35% der Gesamtnote**
- Klausur am Semesterende (60min)
 - **50% der Gesamtnote**

Anmeldefristen

- ISIS für Vorlesungsmaterial – **am besten sofort**
- Modulanmeldefrist via Moses oder Prüfungsamt
- Für Portfolioprüfung
 - 16.10.2023 bis 05.11.2023
 - Abmeldung bis spätestens 09.11.2023
 - Empfehlung: Wählen Sie den ersten Termin am 04.03.2024
- Bereits zur Prüfung Zugelassene dürfen die Prüfung in der (alten) schriftlichen Form ablegen:
 - 90 min Klausur am 04.03.2024
 - 16.10.2023 bis 05.11.2023
 - Abmeldung bis spätestens 04.02.2024
- **Beachten Sie die Ankündigungen in ISIS !**

Prüfungsmodalitäten

- Portfolioprüfung
 - **Programmierkurs (15%)**
+ Programmieraufgaben (35%)
+ Klausur (50%)
 - Schriftlicher Test: 04.03.2024 08:00 – 10:30 Uhr
 - Wiederholungsmöglichkeit: 27.03.2024 08:00 – 10:30 Uhr

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 vom Programmierkurs. Ausgabe: 16-27.10.2023, Abgabe: 10.11.2023
- **Programmieraufgaben (35%)**
 - Kategorie grün – Ausgabe: 02.11.2023, Abgabe: 17.11.2023
 - Kategorie blau – Ausgabe: 23.11.2023, Abgabe: 08.12.2023
 - Kategorie rot – Ausgabe: 07.12.2023, Abgabe: 09.01.2024
 - Kategorie gold – Ausgabe: 11.01.2024, Abgabe: 02.02.2024

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/>

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

VL 7 „Teile und Herrsche I“: Einführung der algorithmischen Methode, Merge Sort

VL 8 „Korrektheitsbeweise“: Rechnermodell, Beispielbeweise

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