

Algorithmen und Datenstrukturen

0. Start und Organisatorisches zur Vorlesung

Prof. Dr.-Ing. Marc Stamminger



Algorithmen? Kinderleicht!



Quelle: http://www.youtube.com/watch?v=cVMKXKoGu_Y

Als Programmcode

```
public int[] sort(int l, int r) {  
    int q;  
    if (l < r) {  
        q = partition(l, r);  
        sort(l, q);  
        sort(q + 1, r);  
    }  
    return intArr;  
}
```

<https://javabeginners.de>

```
int partition(int l, int r) {  
    int i, j, x = intArr[(l + r) / 2];  
    i = l - 1;  
    j = r + 1;  
    while (true) {  
        do {  
            i++;  
        } while (intArr[i] < x);  
  
        do {  
            j--;  
        } while (intArr[j] > x);  
  
        if (i < j) {  
            int k = intArr[i];  
            intArr[i] = intArr[j];  
            intArr[j] = k;  
        } else  
            return j;  
    }  
}
```

Vorlesung „Algorithmen und Datenstrukturen“

■ Dozent:

- Prof. Dr.-Ing. Marc Stamminger
Lehrstuhl Graphische Datenverarbeitung
 - `marc.stamminger@fau.de`
 - `http://lgdv.tf.fau.de/`



■ Termine: (bitte kommen Sie *pünktlich*)

~~□ Dienstag, 08:15 - 09:45,~~

~~□ Donnerstag, 16:15 - 17:45,~~

Optionales Live-Webinar mit Q&A am

□ Dienstag, 08:30 - 10:00,

~~Hörsaal H~~

~~Hörsaal H~~

Vorlesungs-
aufzeichnungen
über StudOn

Zoom-Link siehe StudOn

Vorlesung und Übung im Web (I)

- Adresse:

- <https://www.studon.fau.de/crs3063592.html>

- Beitritt mit Kurspasswort:



- Dort finden Sie:

- **Organisatorische Hinweise und Ankündigungen:**

- Aktuelle Informationen (**RSS**-Feed abonnieren! Dringendes tlw. auch per E-Mail, bitte beachten!), Literaturempfehlungen, Prüfungsmodalitäten, zu verwendende Software und Hinweise zu ihrer Einrichtung, alte Klausuren und alte Übungsblätter früherer Semester, Evaluationsergebnisse, **FAQ** - Häufig gestellte Fragen und Antworten

Vorlesung und Übung im Web (II)

■ Adresse:

□ <https://www.studon.fau.de/crs3063592.html>

Kurspasswort:
AUDFAU

□ **Vorlesung:** Gliederung, Folien, Zusatzmaterial, ...

Die Vorlesungsfolien werden dort i. d. R. bis spätestens am Tag vor der jeweiligen Vorlesung bereitgestellt/aktualisiert.
PDF-Format in zwei Varianten: schwarz-weiß und farbig.
Änderungen (neue PDFs) im Laufe des Semesters möglich!

□ **Übung:** Aufgabenblätter und Folien für die Tafelübungen, ggf. Zusatzmaterialien, Abgabetermine für die Lösungen, Übungskoordinatoren und -tutoren, Hinweise zur sog. Intensivübung für Programmieranfänger

□ **Abgabe von Hausaufgaben je nach Aufgabe:**
entweder **StudOn** oder **EST (Exercise Submission Tool)**

- Modul „Algorithmen und Datenstrukturen“ **bestanden**, wenn

- ☐ **(unbenoteter) Übungsschein** erhalten
und

- ☐ **(benotete) Klausur** bestanden



- Reihenfolge egal.

- Bei Nichtbestehen eines Teils kann dieser im Folgesemester wiederholt werden.

- Jede erbrachte Teilleistung bleibt unbegrenzt erhalten.

- **Klausur**

- ☐ Dauer: 120 Minuten

- ☐ **Voraussichtlicher Termin: ca. 29. März 2021**

(Achtung: Änderungen auch kurzfristig möglich, daher unbedingt regelmäßig Ankündigungen beachten!)

- 1 mal pro Woche: Ihr **Übungsblatt**
 - Im Laufe des **Freitags** erscheint ein Übungsblatt mit Theorie- und Programmieraufgaben auf der StudOn-Seite.
Zweck: Praktische Auseinandersetzung mit dem Vorlesungsstoff.
 - 10 Tage Bearbeitungszeit,
Montags Abgabe zur Korrektur/Bewertung.
 - Die Aufgaben sind
 - Gruppenaufgaben, die im **Zweierteam** gelöst werden sollen, und
 - **individuell** zu lösende Einzelaufgaben.
 - Pro Übungsblatt gibt es i. d. R. ca. 60 Punkte.
Voraussichtlich gibt es insgesamt 12+1 Übungsblätter.
 - Insgesamt entfallen jeweils ca. 50% der erreichbaren Punkte auf Einzel- bzw. Gruppenaufgaben, wobei am Semesterbeginn mehr Gruppenaufgaben und am Semesterende mehr Einzelaufgaben gestellt werden.
 - **Unbenoteten Übungsschein** gibt's, wenn **sowohl 60% der möglichen Einzelpunkte als auch 60% der Gruppenpunkte** erreicht wurden.

AuD-Übungsbetrieb (II)

- 1 mal pro Woche: Ihre **Tafelübung**/Ihr **Tutorium** (2 SWS)
 - Klärung von Fragen zum Vorlesungsstoff und zum nächsten Übungsblatt, Besprechung der Lösung des letzten Übungsblattes.
 - Verschiedene Termine zur Auswahl
 - **Anmeldung über StudOn zwingend erforderlich!**
 - Eine **Schritt-für-Schritt-Anleitung zur Verwendung von StudOn und EST** im Übungsbetrieb finden Sie im AuD-StudOn-Kurs!
- Nach Bedarf: Ihre betreute **Rechnerübung** in den ~~CIP-Pools~~

b.a.w. online!

 - Bearbeitung der Praxishausaufgaben aus den Tafelübungen, Klärung von technischen oder Java- bzw. Programmier-Problemen mit dem jeweils anwesenden Tutor (nach entsprechender Vorbereitung!)
 - Wählen Sie einen (beliebigen) Rechnerübungstermin aus. Bei Bedarf können Sie mehrere Rechnerübungen besuchen.
 - Rechner im CIP-Pool stehen allen Studierenden zur Verfügung. AuD-Studenten haben an den Rechnerübungsterminen Priorität.
- **Tafel-/Rechnerübungen starten regulär am: 09.11.2020**

- Nach Bedarf: **Intensivübungen**
 - Freiwillige zusätzliche Veranstaltung
 - insbesondere für Programmieranfänger/-innen
 - späterer Einstieg möglich, keine Anwesenheitspflicht
 - nicht noten- oder scheinrelevant
 - Inhalt und Ablauf:
 - Teilnehmer schlagen Themen/Fragen/Aufgaben/... via StudOn vor
 - Antwort/Lösung wird wie in einem „Workshop“ gemeinsam erarbeitet
 - Programmierung wird an Beispielen live vorgeführt
 - **Termin/Raum-Findung per Umfrage via StudOn**
 - Ankündigung in StudOn beachten

Anmeldung zum Übungsbetrieb/Tafelübungen – Termine

- Anmeldung zu einer Tafelübung über StudOn
in der ersten Vorlesungswoche zwingend erforderlich,
Angaben zu Wunschterminen/Verfügbarkeiten
bis spätestens Freitag Nacht aktualisieren!
- Sie werden am darauffolgenden Wochenende automatisch und **verbindlich** für eine der wöchentlichen Tafelübungen eingeteilt.
- Keine Anmeldung zu Rechnerübungen notwendig.
- Übungsstart ist am 09.11.2020

■ Übungscoordination

▼ Dozenten

Organisation



**Norbert
Oster**

■ eMail-Kontakt:

□ cs2-aud@fau.de

Abgabe der Lösungen – Allgemeines

- **Ausschließlich elektronisch** über StudOn bzw. EST.
- Die **Abgabefrist endet** i.d.R. jeweils 10 Tage nach der Ausgabe des Übungsblatts - am **Montag um 9:59 Uhr**.
- Laden Sie Ihre Lösung möglichst nicht in der letzten Minute hoch, damit Sie nicht an eventuell auftretenden Rechner/Netz-Problemen scheitert.
- **Nachträgliche Abgaben sind nicht möglich!**
- Nach der Abgabe erfolgt Korrektur und Bewertung teilautomatisch und durch Ihre/-n jeweilige/-n Tutor/-in.
- Korrigierte Abgabe und Anmerkungen zur Bewertung stellen wir wieder im StudOn bzw. EST zur Einsicht bereit.

Mehrmalige Abgabe innerhalb der Frist ist möglich, **letzte** Version gilt!

Abgabe der Lösungen – Gruppenaufgaben

- Die als Gruppenaufgaben gekennzeichneten Hausaufgaben können und sollten Sie mit einem Partner bearbeiten.
- Diese/n Partner/in können Sie bei Bedarf von Aufgabe zu Aufgabe wechseln (z. B. bei Problemen in der Kooperation).
- Es ist bei Bedarf auch möglich, die Gruppenaufgaben alleine zu bearbeiten.
- Die technischen Details hierzu erfahren Sie in der Übung.
- Sollten Sie einen neuen Partner benötigen, nutzen Sie z. B. das AuD-Forum im StudOn-Kurs für Ihre Anfrage.

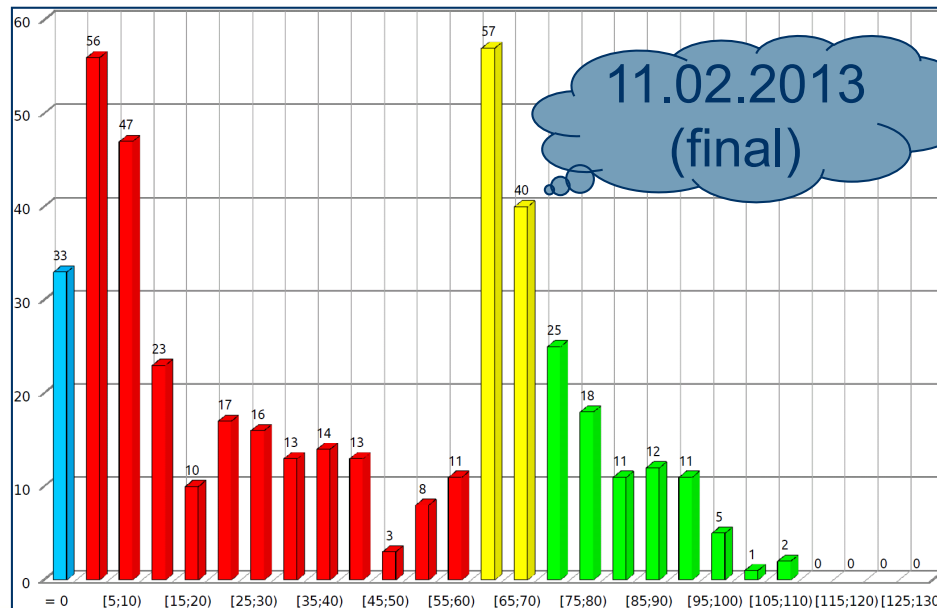
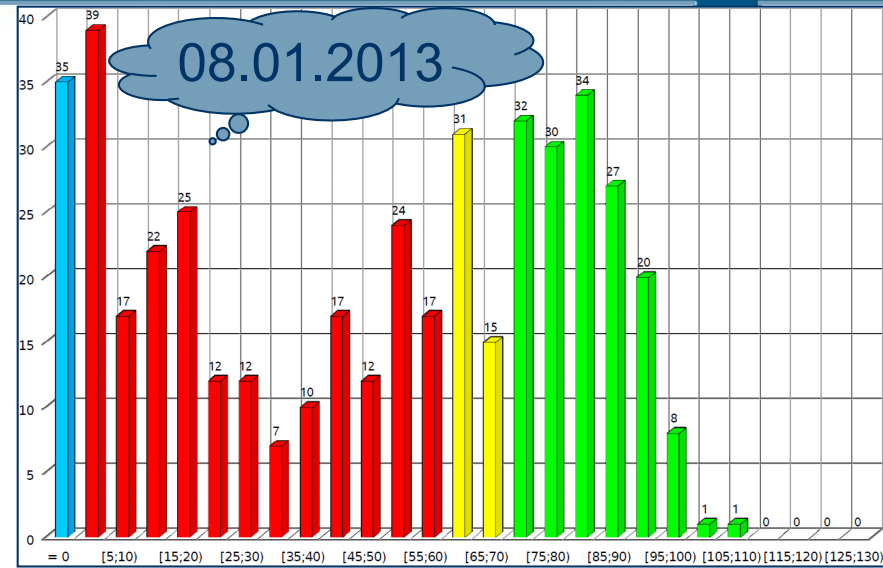
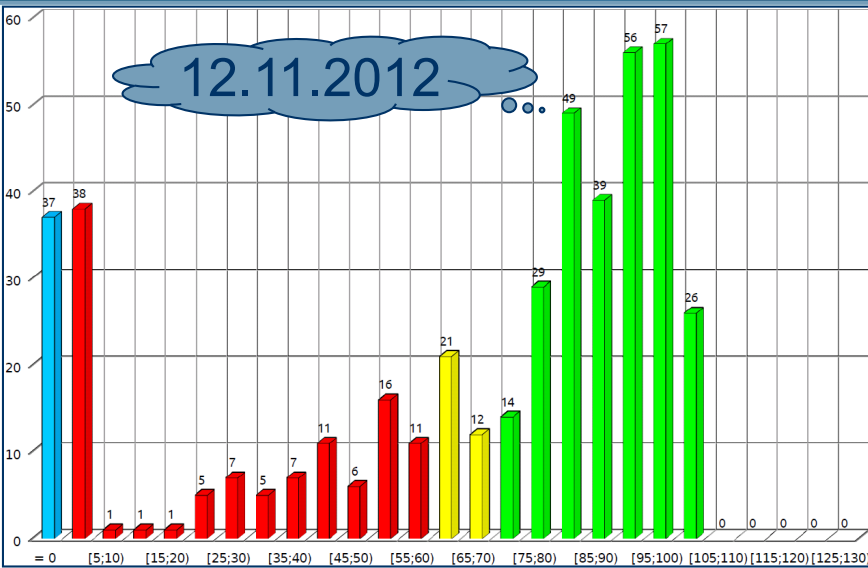
Plagiate

- Wir nutzen eine Software, die alle Einzellösungen sowie alle Gruppenlösungen untereinander vergleicht und zu ähnliche Lösungen als Plagiate identifiziert.
- Im Verdachtsfall werden **beide** Lösungen mit **0 Punkten** bewertet.
- Bitte unbedingt die **wichtigen Hinweise** auf dem organisatorischen „**Übungsblatt 0**“ beachten.

- Wie viel Zeit müssen Sie für die LV „Algorithmen und Datenstrukturen“ investieren, um erfolgreich zu sein?
 - Der LV sind 10 Leistungspunkte (LP) zugeordnet, das entspricht **300 Std.** verteilt auf das gesamte Semester.
 - **Präsenzveranstaltungen:**
 - (2 Vorlesungen pro Woche (2 x 90 min = 3 h)
 - + 1 Tafelübung pro Woche (= 1,5 h)
 -) x 15 Semesterwochen
 - = **68 Std.**
 - **Eigenarbeit: 232 Std.**
- Rechnen Sie mit 2 Tagen / Woche in der Vorlesungszeit

- Programmieren lernt man nicht durch Zuschauen, sondern durch **ÜBEN**
 - Sie werden dazu in AuD viele Möglichkeiten bekommen
 - Nutzen Sie diese Möglichkeiten !
 - Machen Sie die Aufgaben auch selbst !
- Durchfallquoten relativ hoch
 - Investieren Sie Zeit !
 - Wir können Sie nur begleiten !
- Manche von Ihnen hatten schon vieles aus AuD in der Schule, Ihr Vorsprung wird aber schnell schmelzen 😬

Übungsergebnisse zu verschiedenen Terminen am Beispiel WS 2012/13



Übungsschein (WS12/13)

Angemeldet: 413

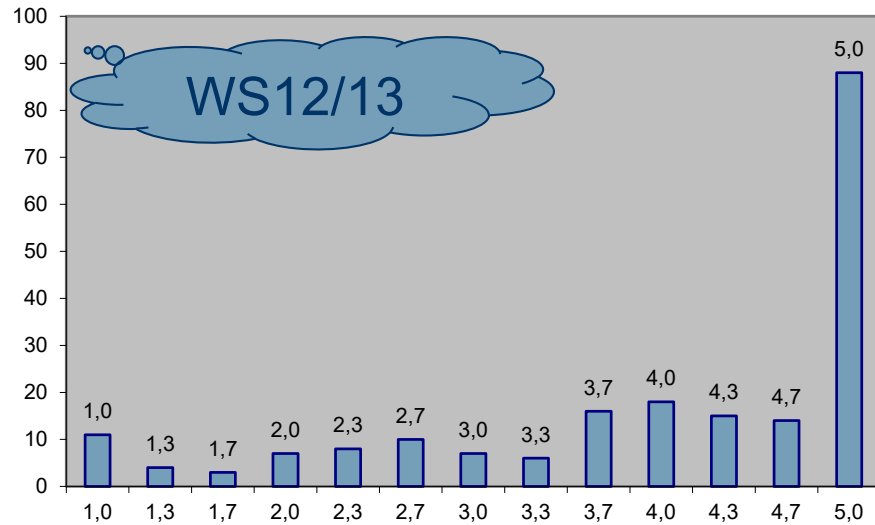
181 x bestanden,

232 x nicht bestanden

Durchfallquote: 56%

Klausurergebnisse früherer Semester

Notenverteilung



Klausur (WS12/13)

90 x bestanden; 117 x nicht bestanden

Nicht erschienen: 213

Ø Note: 3,98

Durchfallquote: 57%

Klausur (SS2015)

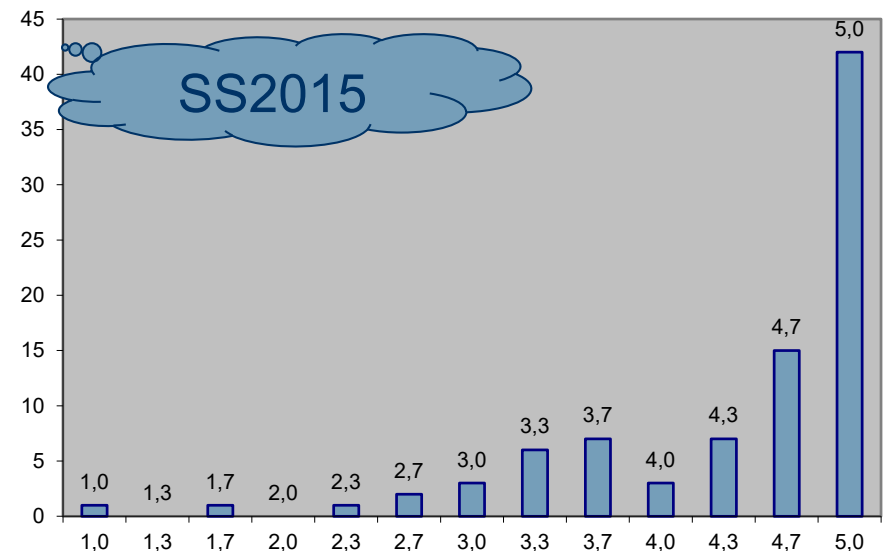
24 x bestanden; 64 x nicht bestanden

Nicht erschienen: 168

Ø Note: 4,41

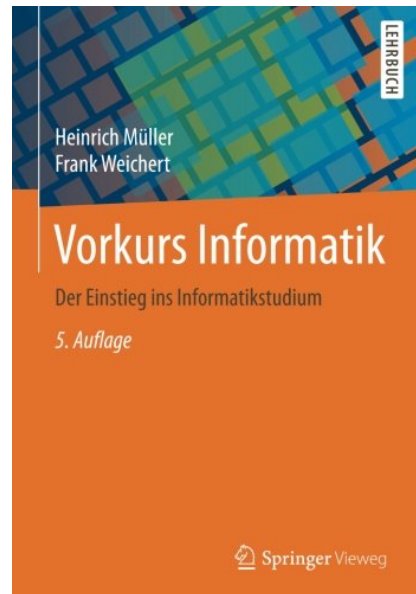
Durchfallquote: 73%

Notenverteilung



Literaturempfehlungen: Einführung für Anfänger/-innen

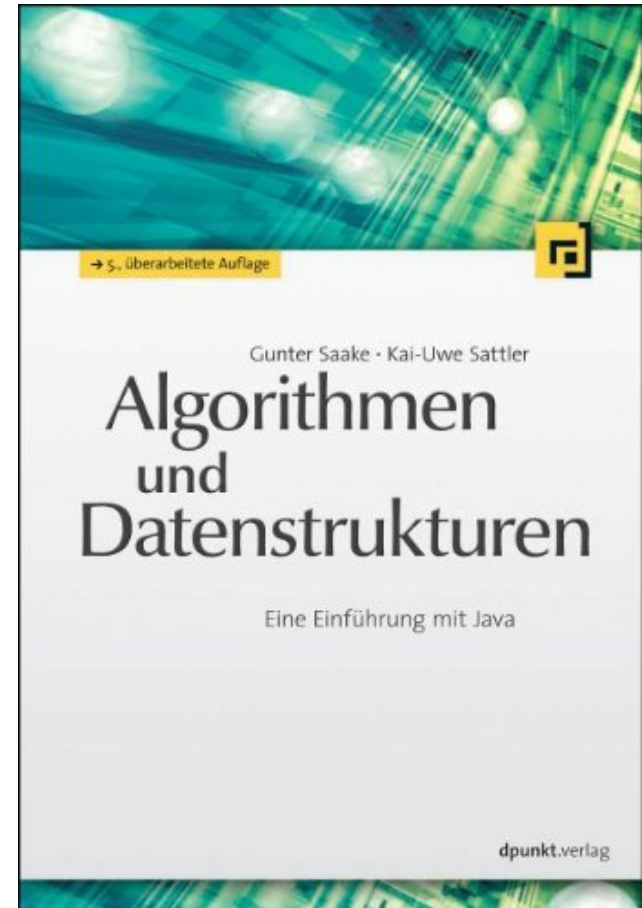
- Heinrich Müller, Frank Weichert: Vorkurs Informatik. Springer Vieweg, ca. 25,- €
 - Leicht lesbare und didaktisch gut aufbereitete Einstiegshilfe für Programmierneulinge
- <http://dx.doi.org/10.1007/978-3-658-16141-5>
kostenlose Download-Version
(innerhalb des Uni-Netzes, auch über OPAC/Uni-Bib erreichbar)



5. Auflage: 2017, 408 S.

Literaturempfehlungen: Standardwerk

- Gunter Saake, Kai-Uwe Sattler: Algorithmen und Datenstrukturen - Eine Einführung mit JAVA. 5. Auflage, dpunkt.verlag, 2013, 576 S.
- Zweck:
 - Grundlagen- und Vertiefungswerk zum Inhalt der Lehrveranstaltung
 - Java-Programmierung
 - Algorithmen und Datenstrukturen
- Kosten:
 - ca. 45,- EUR



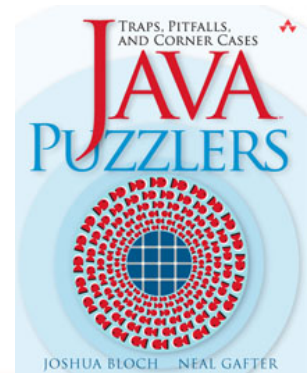
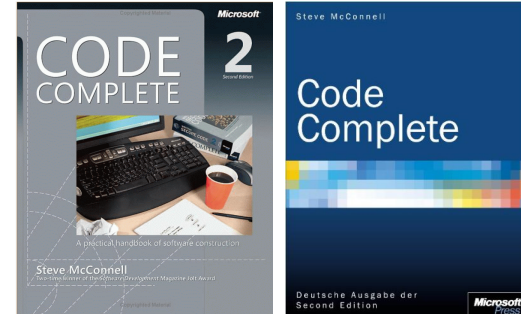
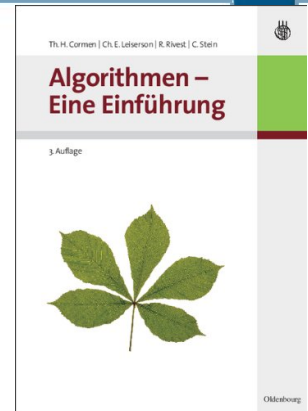
Literaturempfehlungen: Java-Nachschlagewerke

- Christian Ullenboom: Java ist auch eine Insel.
11. Auflage, Galileo Computing, 2014, 1306 S.
<http://openbook.galileocomputing.de/javainsel/>
Print-Version: ca. 50 EUR, kostenlose Online/Download-Version
- Guido Krüger, Heiko Hansen: Handbuch der JAVA-Programmierung. Addison-Wesley, 2011, 1408 S.
<http://www.javabuch.de/>
Printversion: ca. 50 EUR, kostenlose Download-Version
- Zweck:
 - Umfassende sehr detailliertes Nachschlagewerke zur Java-Programmierung
 - Hervorragend geeignet, um zu speziellen Programmiertechniken detaillierte Informationen zu bekommen.
 - Gehen beide weit über den Vorlesungsstoff hinaus.



Literaturempfehlungen: Weiterführende Werke

- Cormen, Leiserson, Rivest, Stein:
Algorithmen – Eine Einführung. Oldenburg Verlag, 2010
 - Vertiefungswerk zum Inhalt der Vorlesung
 - Kosten: ca. 80 EUR (1339 S.)
- Steve McConnell: Code Complete – A Practical Handbook of Software Construction. Microsoft Press, 2004
 - praxisorientierte Darstellung von Softwareentwicklung
 - Kosten: ca. 35 EUR (960 S.)
- Joshua Bloch, Neal Gafter: Java Puzzlers: Traps, Pitfalls, and Corner Cases. Addison-Wesley, 2005
 - Kosten: ca. 30 EUR (282 S.)
 - Code-Beispiele: <http://www.javapuzzlers.com/>



... und jetzt geht es richtig los!

Me and the boys ready for Zoom

