

Führen Sie die folgenden Übungen an einem der Laborrechner oder Ihrem eigenen Rechner aus. Nutzen Sie die Eclipse-IDE für die Entwicklung mit der Programmiersprache JAVA.

1. Grundlagen und Implementierung von Hashverfahren
 - (a) Wozu dienen Hashverfahren (Nennen Sie Anwendungsgebiete)?
 - (b) Welche Vor- und Nachteile bieten Sie für welche Anwendungsgebiete?
 - (c) Was versteht man unter dem Horner-Schema und wofür wird es beim Hashing nutzbar?
 - (d) Implementieren Sie notwendige Funktionen zur Verwaltung einer Hashtabelle für einfache Zahlenwerte.
2. Implementierung von Sondierungsfunktionen
 - (a) Wann und wozu werden Sondierungsfunktionen gebraucht?
 - (b) Welche Sondierungsfunktionen sind Ihnen bekannt (mindestens 3)?
 - (c) Üben Sie anhand einiger Beispiele den Umgang mit verschiedenen Sondierungsfunktionen.
 - (d) Implementieren Sie eine der von Ihnen genannten Sondierungsfunktionen und leiten Sie deren Komplexität im Worst-, Average- und Best-Case her.
3. Implementierung einiger Sortierfunktionen
 - (a) Implementieren Sie die Datenstruktur aus Übung 2 als Array.
 - (b) Implementieren Sie 4 verschiedene Sortieralgorithmen zur Sortierung des Arrays anhand der Matrikelnummer.
 - (c) Wie müssten Sie ihre Sortierfunktionen verallgemeinern, damit ohne „Copy+Paste“ auch nach den anderen Merkmalen der Datenstruktur aus Übung 2 sortiert werden kann? (Stichpunkte zur JAVA-Implementierung - nicht direkt implementieren)

Hinweise: Zum Lösen der Aufgabe verwenden Sie bitte die Standarddatentypen und -definition von JAVA. Sollten Sie diese Übung abgeben wollen, geschieht das bitte in Form eines Archives mit einer Lösungsbeschreibung im PDF-Format (sowie Lösungen der Aufgaben, die nicht die Implementierung betreffen) und einem Ordner mit dokumentiertem Quelltext (Javadoc). Zusätzlich soll der Anwender die Möglichkeit haben die Anwendung per Console zu bedienen. Sollten Sie mit noch keinem Textverarbeitungsprogramm vertraut genug sein, sei Ihnen ~~ME~~ X wärmstens ans Herz gelegt.

Deadline und Punkte

Abgabetermin: 23.1.2017 - 23:55 Uhr.

Maximale Punktzahl: 3

Wichtige Informationen zur Abgabe

Makefile: Es muss bei jeder Abgabe eine **Makefile** im Rootverzeichnis vorhanden sein. Ohne die Makefile wird **keine Bewertung der Aufgabe** vorgenommen. Es dürfen in dieser nur **relative Pfade** verwendet werden. Zusätzlich müssen in dieser die folgenden Targets definiert und ausführbar sein:

- **default:** Kompilierung der Java-Konsolenanwendung,
- **run-app:** Ausführung der Java-Konsolenanwendung,
- **run-test:** Für die Durchführung der JUnit-Tests (Jede Funktionalität soll gegen einen JUnit-Test getestet werden),
- **doc:** Für die Generierung der Quelltext-Dokumentationen,
- **clean:** Säubert das Arbeitsverzeichnis von den automatisch generierten Dateien (Binärdatei z.B. *.class),

Zusätzliche Literatur: Im Folgenden sind Literaturverweise für das Lösen der Aufgabenblätter angegeben:

- Make¹
- JUnit²
- Java OpenJDK³
- Openbook: Shell-Programmierung⁴
- Openbook: Java ist auch eine Insel⁵
- Pro Git Book - Englisch (PDF etc.)⁶
- Pro Git Book - Deutsch⁷
- HTW Berlin - Git-Server⁸

Verwalten Sie den Source Code über die Versionsverwaltung Git als Übung für zukünftige Projekte in der AI.

Abschließende Informationen Vergessen Sie bei der Entwicklung der Java-Konsolenanwendung die Benutzerfreundlichkeit nicht. Über produktives und sachliches Feedback zu den Aufgabenblättern und Übungen von Ihnen würde ich mich sehr freuen. Viel Spass und Erfolg bei der Entwicklung der Java-Konsolenanwendungen.

¹<https://www.gnu.org/software/make/>

²<http://junit.org/>

³<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

⁴http://openbook.rheinwerk-verlag.de/shell_programmierung/

⁵<http://openbook.rheinwerk-verlag.de/javainsel/>

⁶<https://git-scm.com/book/en/v2>

⁷<https://git-scm.com/book/de/v1>

⁸<https://studi.f4.htw-berlin.de/www/>