

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. Lauflängenkodierung

- (a) Erarbeiten Sie sich die Implementierung in Pseudocode einer Lauflängenkodierung für Groß- und Kleinbuchstaben!
- (b) Welche Stärken und Schwächen hat die Lauflängenkodierung?
- (c) Implementieren Sie die Lauflängenkodierung anhand Ihres selbst formulierten Pseudocodes!
- (d) Implementieren Sie auch eine Funktion zum Dekodieren!

2. Huffman-Kodierung

- (a) Erarbeiten Sie sich für die Programmiersprache C eine sinnvolle Datenstruktur zur Berechnung der Huffman-Kodierung einer Zeichenkette.
- (b) Benennen Sie wichtige Funktionen, die zur weiteren Umsetzung notwendig sind.
- (c) Machen Sie sich mit bitweisen Operationen (Setzen, Clearen, Verschieben und Verknüpfen) in der Programmiersprache JAVA vertraut.
- (d) Implementieren Sie die notwendige(n) Datenstruktur(en) für die Huffman-Kodierung.
- (e) Implementieren Sie die Funktionen zum Kodieren und Dekodieren.

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~~ 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/>