Programmierkurs 2
Sommersemester 2023

3. Übungsblatt (BLOCK 1)

Abgabe bis 15.05.2022

Legen Sie in Ihrem git-Ordner einen Unterordner **blatt3** an. Checken Sie dann den Code für das Übungsblatt dort ein.

Binärer Suchbaum

Implementieren Sie einen binären Suchbaum. Die Datenstruktur soll das Einfügen von neuen Knoten und das Finden von Knoten mit einem bestimmten Schlüssel mit Laufzeiten proportional zur Höhe des Baumes ermöglichen.

Das Ziel ist es, eine Suche in einer Filmdatenbank zu realisieren, in welcher Filmtitel und die zugehörige Kurzzusammenfassung des Filmes stehen. Bei Eingabe des Titels möchte man dann effizient die zugehörige Beschreibung bekommen. Filme und Beschreibungen werden in der Eingabedatei spezifiziert. Dabei besteht jede Zeile aus dem Titel und der zugehörigen Beschreibung, separiert durch einen Tab. Legen Sie eine Filmklasse an und implementieren Sie einen Film-Comparator basierend auf dem Titel. Bauen Sie dann den binären Suchbaum für alle Filme auf durch wiederholtes Aufrufen der 'insert'-Funktion. Fügen Sie dabei die Filme gemäß der Reihenfolge in der Eingabedatei ein.

Lesen Sie dann die zweite Eingabedatei, in welcher nur Filmtitel stehen (wieder einer pro Zeile). Suchen Sie dann nacheinander die entsprechenden Titel in Ihrem Suchbaum mit der 'find'-Methode. Extrahieren Sie für jeden Titel die zugehörige Beschreibung und schreiben Sie diese passend zur Eingabereihenfolge der Titel in die Ausgabedatei (eine Zeile pro Beschreibung).

Speichern Sie Ihren Code in der Datei **Search.java** ab.

COMPILE

```
javac Search.java
```

AUFRUF:

```
java Search <EingabeFile1> <EingabeFile2> <AusgabeFile>
```

TEST RUNS (müssen durchlaufen, korrekte Ausgaben im ILIAS zum Vergleich):

```
java Search filme-mit-beschreibung-klein.txt filmliste.txt beschreibungen.txt
```

FINAL RUNS (sollten in höchstens 5 Minuten durchlaufen und plausible Ergebnisse liefern):

```
java Search filme-mit-beschreibung-gross.txt filmliste.txt beschreibungen.txt
```