Programmiertechnik II
Angewandte Informatik
SS 2018
Prof. Dr. Oliver Bittel

HOCHSCHULE KONSTANZ
TECHNIK, WIRTSCHAFT UND GESTALTUNG

Aufgabenblatt 12 – Collection-Typen und Lambda-Ausdrücke

Aufgabe 1

Schreiben Sie eine Funktion, die für eine Datei die Häufigkeiten der in ihr vorkommenden Wörter bestimmt. Verwenden Sie dazu den Container-Typ **java.util.map**. Geben Sie dann die 100 häufigsten Wörter aus.

Auf der Web-Seite finden Sie ein Java-Programm, das alle Wörter eines deutschsprachigen Textes einliest und ausgibt (Leerraum-, Satzzeichen und andere Sonderzeichen werden als Trennzeichen interpretiert). Ergänzen Sie die beiden Methoden geeignet:

- ermittleHaeufigkeiten
- printTop100

Hinweis: die Methode printTop100 lässt sich besonders elegant und effizient mit Hilfe eines Streams implementieren. Die Methode kann aber auch „konventionell“ mit Hilfe einer Liste implementiert werden.

Wählen Sie aus

http://www.digbib.org/

ein beliebiges elektronisches Buch (Roman oder Erzählung) im Text-Format aus und entfernen Sie die DigBib.Org-Hinweise am Anfang der Datei. Führen Sie dann eine Häufigkeitsanalyse durch.

Aufgabe 2

Machen Sie sich mit der Klasse **LocalDate** aus der **Java API** vertraut. Schreiben Sie eine Klasse Person bestehend aus Vorname, Nachname und Geburtsdatum vom Typ LocalDate. Versehen Sie diese Klasse mit einem Konstruktor, entsprechende setter/getter-Methoden und einer toString()-Methode. Schreiben Sie eine main-Methode, die eine kleine Liste persList mit Personen anlegt und dann folgende Aufgabenstellungen löst:

a) Definieren Sie ein Prädikat als Lambda-Ausdruck, das prüft ob eine Person volljährig ist, und prüfen Sie mit Hilfe dieses Prädikats in einer Schleife, ob alle Personen der Liste persList volljährig sind.

b) Sortieren Sie persList aufsteigend nach dem Geburtsdatum. Verwenden Sie die Methode Collections.sort(list, cmp), indem ein Lambda-Ausdruck cmp vom Typ Comparator<Person> übergeben wird.

c) Sortieren Sie persList absteigend nach dem Geburtsdatum. Ändern Sie dazu den Comparator aus b) mit Hilfe der reversed-Methode ab (siehe Java-API zu Comparator).

d) Erzeugen Sie aus der Liste persList einen Strom und führen Sie folgende Strom-Operationen durch:

- alle volljährigen Personen eliminieren. Verwenden Sie die filter-Methode und das Volljährigkeits-Prädikat aus a) und negieren es mit Hilfe von negate (siehe Java API Predicate).
- die Geburtsdaten extrahieren,
- sortieren und
- ausgeben.

e) Erzeugen Sie aus der Liste persList einen Strom und führen Sie folgende Strom-Operationen durch:

- jeden Familienname in Großbuchstaben ändern,
- nach dem Namen sortieren und
- ausgeben.