

## Programmierung 2 - Sommersemester 2022

Prof. Dr.-Ing. Martin Burger

### Übungsblatt Nr. 21 Abgabe KW 27

**Allgemeiner Hinweis:** Implementieren Sie für alle Ihre Lösungen einen Dialog-Test, auch wenn dies bei einzelnen Aufgaben nicht explizit in der Aufgabenstellung genannt ist. Der Test sollte so gestaltet sein, dass die Funktionsweise Ihres Programms bei der Abnahme gut nachvollzogen werden kann.

#### 1. Aufgabe

Implementieren Sie die Methoden, welche der `Lager`-Klasse in Übungsblatt 18 hinzugefügt wurden mithilfe von Streams neu. Ggf. ist es sinnvoll die an die Methoden übergebenen Parameter und den Returntypen anzupassen, um besser mit den Streams arbeiten zu können.

#### 2. Aufgabe

Schreiben Sie ein C-Programm, das vom Benutzer zunächst die Eingabe eines Geldbetrags (Nettopreis) erwartet, welcher mit `scanf()` eingelesen wird. Es sollen dann 20% Mehrwertsteuer hinzugezogen (ergibt den Bruttobetrag) und schließlich 2% Skonto abgezogen (ergibt den Rechnungsbetrag) werden. Die Beträge sollen dann in folgendem Format ausgegeben werden:

```
Nettopreis           Euro xxxx.xx
+ 20% MwSt           Euro xxxx.xx
=====
Bruttobetrag         Euro xxxx.xx
- 2% Skonto          Euro xxxx.xx
=====
Rechnungsbetrag      Euro xxxx.xx
```

#### 3. Aufgabe

Implementieren Sie ein C-Programm, welches eine Zahl `n` vom Nutzer einliest und dann iterativ die ersten `n` Fibonacci-Zahlen berechnet und auf dem Bildschirm ausgibt.

#### 4. Aufgabe (Zusatzaufgabe)

Schreiben Sie ein C-Programm, welches für die Typen `char`, `short`, `int`, `unsigned char`, `unsigned short` und `unsigned int` den maximalen und den minimalen Wert feststellt und ausgibt. Geben Sie zum Vergleich auch die Werte der Bibliothek `limits.h` der C Standard Library aus.

#### 5. Aufgabe (Zusatzaufgabe)

Die Übergabe von Parametern an eine Funktion erfolgt in C nach dem call-by-value-Prinzip. Schreiben Sie ein Programm welches dies nachweist.

6. Aufgabe (**Zusatzaufgabe**)

- (a) Implementieren Sie folgende Methode:

```
public List<String> readTextFile(String file, String prefix, int minLength, int limit)
```

Die Methode soll die gegebene Textdatei `file` einlesen. Filtern Sie anschließend die ersten `limit` unterschiedlichen Wörter heraus, die mit dem Substring `prefix` beginnen und mindestens eine Länge von `minLength` haben. Sortieren Sie die resultierenden Wörter alphabetisch und geben Sie das Ergebnis als Liste zurück.

**Wichtig:** Implementieren Sie Ihre Lösung mithilfe eines Streams!

Testen Sie Ihre Lösung mit dem RFC 791 als Input: <https://tools.ietf.org/rfc/rfc791.txt>

- (b) Implementieren Sie eine zweite Methode analog zur Methode `readTextFile` mit dem Unterschied, dass Sie eine Map zurückgeben, welche die resultierenden Wörter nach ihrer Länge gruppiert enthält.