

## Dateien

Schreiben Sie ein Programm

- a) in Java
- b) in C oder in C++,

das Variablen folgender Typen

- Zeichen
- Integer-Zahl
- Float-Zahl
- String

auf eine

- a) Textdatei
- b) binäre Datei

schreibt.

Das Programm soll die beiden Dateitypen wieder lesen und die Werte in andere Variablen als die ursprünglichen ablegen können. Darüber hinaus sollen die beiden Dateitypen byteweise gelesen werden und die einzelnen Bytes in dezimaler, hexadezimaler, binärer und Zeichenform auf die Konsole ausgegeben werden können.

Der Ablauf ist mittels eines einfachen Kommandoprozessor gesteuert, z.B. so:

Menue

```

auf Textdatei schreiben:   WT
aus Textdatei lesen:      RT
auf Binärdatei schreiben: WB
aus Binärdatei lesen:     RB
Datei byteweise ausgeben  B
Quit - Programmende      Q
  
```

Der Dump kann z.B. bei einer Textdatei wie folgt aussehen:

```

Addr  Dec   Hex   Bin      ASC
0000:  97    61    01100001  'a'
0001:  13    0d    00001101  '.'
0002:  10    0a    00001010  '.'
0003:  49    31    00110001  '1'
0004:  50    32    00110010  '2'
0005:  13    0d    00001101  '.'
0006:  10    0a    00001010  '.'
0007:  49    31    00110001  '1'
0008:  50    32    00110010  '2'
0009:  46    2e    00101110  '.'
0010:  48    30    00110000  '0'
0011:  13    0d    00001101  '.'
0012:  10    0a    00001010  '.'
0013: 104    68    01101000  'h'
0014:  97    61    01100001  'a'
0015: 108    6c    01101100  'l'
0016: 108    6c    01101100  'l'
0017: 111    6f    01101111  'o'
0018:  13    0d    00001101  '.'
0019:  10    0a    00001010  '.'
  
```

Bei der binären Datei prüfen Sie rechnerisch, ob die Werte der einzelnen Bytes Ihren Erwartungen entsprechen.

Empfehlung:

In Java können Sie die Methode `System.out.format` der Klasse `java.io.PrintStream`