

Zeichen in Dateien

In diesem Abschnitt soll der Umgang mit einfachen **Textdateien** behandelt werden. Textdateien haben eine relativ einfache Struktur. Sie bestehen aus Zeilen, die mit Zeilenendezeichen abgeschlossen werden. Das ist das Zeichen LF (Zeilenvorschub; engl.: line feed) oder eine Kombination der Zeichen LF und CR (Wagenrücklauf; engl.: carriage return). Unter den Betriebssystemen Unix, Linux und Mac-OS wird das Zeilenende durch LF und unter Windows durch CR-LF markiert. Das Ende einer Textdatei wird oft durch ein Zeichen für das Dateiende (EOF oder EOM: ASCII 25; Mit Tastatur: CTRL-Z) festgelegt.

Die folgende Methode demonstriert die Behandlung von Textdateien. Grundlage des Beispiels und der folgenden Aufgaben sollen die Klassen *In* und *Out* sein. Mit `ch = In.read()`; wird ein Zeichen gelesen. Wenn mit `In.open(dateiname)`; keine Datei angegeben wird, erfolgt die Eingabe von der Konsole. Hier wird also mit `In.open(dateiname)`; auf die Eingabe aus einer Datei umgeschaltet. Ebenso erfolgt die Ausgabe mit `Out.print(ch)`; auf die Konsole, wenn nicht mit `Out.open(dateiname)`; eine Datei angegeben wird. Beim Arbeiten mit Dateien muss jeweils am Ende ein `In.close()`; oder ein `Out.close()`; aufgerufen werden.

Vor der Anwendung lege man zum Testen im aktuellen Projektverzeichnis eine Textdatei mit beliebigem Inhalt an.

```
1  public void dateiAnzeigen(String dateiname)
2  { char ch;
3    boolean okay;
4    In.open(dateiname);
5    if (In.done())
6        { do
7          { ch = In.read();
8            okay = In.done() || (ch == In.eof);
9            if (okay && (ch != In.eof))
10               Out.print(ch);
11          } while ((ch != In.eof) && okay);
12          if (!okay) Out.println("Lesefehler!");
13          In.close();
14        }
15    else { Out.println("Die□Datei□" + dateiname +
16              "□existiert□nicht."); }
17 }
```

Aufgaben:

1. Entwickle ein Programm, das im Dialog (SimpleInput) einen Dateinamen erfragt und dann mit der oben angegebenen Methode eine Datei auf die Konsole ausgibt.

2. Entwickle ebenso ein Programm, das eine Textdatei in eine andere Datei kopiert.
3. Entwickle ein Programm, das eine Textdatei einliest, alle Großbuchstaben in Kleinbuchstaben umwandelt und das Ergebnis in eine andere Datei schreibt.
4. Entwickle ein Programm, das eine Textdatei einliest, die Anzahl der Vokale zählt und das Ergebnis auf der Konsole ausgibt.
5. Entwickle ein Programm, das eine Textdatei einliest, alle Vokale nach dem Motto „Drii Chinisin mit dim Kintribiss“ durch ein „i“ ersetzt und in eine neue Datei schreibt.
6. In der Kryptographie spielt die Häufigkeitsanalyse für Buchstaben in einem Text eine große Rolle. Analysiere den Text in einer Datei und bestimme die prozentualen Häufigkeiten der auftretenden Buchstaben. Informiere dich vorher über die Datenstruktur *array*.
7. Ein einfaches Verfahren zum Verschlüsseln von Texten ist von JULIUS CÄSAR überliefert. Dabei wird jeder Buchstabe um einen bestimmten Wert in der Zeichentabelle verschoben. Ein A wird z.B. ein D, ein B wird ein E usw. Entwickle Programme zum Ver- und Entschlüsseln von Texten in Textdateien.