Textdateien lesen und schreiben

Alle Daten auf dem Computer, die beim Ausschalten nicht verloren gehen sollen, müssen irgendwo gespeichert werden - in der Regel in Dateien: Microsoft Word speichert Dokumente in **`.doc`**-Dateien, Musikstücke werden oft als **`.mp3`**-Dateien gespeichert usw.. Aus einem Java-Programm kann man auf diese Dateien zugreifen. Da aber Daten völlig unterschiedlich abgespeichert werden (eine Word-Datei ist eben kein mp3-Musikstück), ist das Thema sehr umfangreich. Wir begnügen uns an dieser Stelle deswegen mit reinen **`.txt`**-Textdateien, die mit einem einfachen Editor geöffnet werden können. Der folgende Programmabschnitt (zusammen mit **`import java.io.*;`**) liest eine Datei namens **`eingabe.txt`** (welche im Programmverzeichnis existieren muss) und schreibt eine Datei **`ausgabe.txt`**:

```
01 try {
02   // ************* Lesen **************
03   BufferedReader reader = new BufferedReader
04               (new FileReader("eingabe.txt"));
05   String line = null;
06   while ((line = reader.readLine()) != null) {
07     System.out.println("Zeileninhalt: "+line);
08   }
09   reader.close();
10   // ************* Schreiben **************
11   BufferedWriter writer = new BufferedWriter
12                   (new FileWriter("ausgabe.txt"));
13   String[] output = {"Ein","kleiner","Test"};
14   for (int i = 0; i<output.length; i++) {
15     writer.write(output[i]);
16     writer.newLine();
17   }
18   writer.close();
19 } catch (IOException e) {
20   // ... Fehlerbehandlung
21 }
```

Bemerkungen:

Z01, Z19, Z20, Z21: Bei Dateioperationen kann viel schief gehen (z.B. Datei existiert nicht oder schreibgeschützt). Deswegen müssen diese in einem **`try`** ... **`catch`**-Block stehen: Exceptions dienen der eleganten Fehlerbehandlung.

Z03, Z04, Z11, Z12: Das Anlegen der Dateiobjekte ist deswegen so unübersichtlich, da z.B. `mp3`-Dateien ganz anders geöffnet werden müssen, aber dennoch ein Schema eingehalten wird.

Z09, Z18: Werden Dateien am Ende nicht geschlossen, droht ggf. Datenverlust.