

```

// IO.java
// Java Codebeispiel zur Ein-/Ausgabe mit Streams
import java.io.*;

public class IO {
    public static void main(String args[]) throws IOException {
// -----
// 1.a) Zeilenweises Lesen
        DataInputStream inDataStream =
            new DataInputStream(
                new BufferedInputStream(
                    // 1) aus Datei:
                    new FileInputStream("mytext.txt")
                    // 2) aus einem String
                    // new StringBufferInputStream(
                    //     new String("Die ist ein Text\nmit 2 Zeilen."))
                    // 3) von der Standardeingabe (i.d.R. Tastatur)
                    // System.in
                )
            );

        String tmp, text = new String();
        while((tmp = inDataStream.readLine()) != null) {
            // Text zeilenweise einlesen und in Var. "text" speichern.
            // Bem. DataInputStream bietet auch komplexere Funktionen
            // als readLine() an, z.Bsp. readInt(), readBoolean()
            text = text + tmp + "\n";
        }
        System.out.println("1. \n" + text);
        inDataStream.close();

// -----

// 1.b) Byteweises Lesen aus Datei
        BufferedInputStream inBufStream =
            new BufferedInputStream(
                new FileInputStream("mytext.txt"));

        String text1b = new String("");
        byte teil[] = new byte[256];
        int anzGeleseneBytes = 0;
        while(inBufStream.available() > 0) {
            anzGeleseneBytes = inBufStream.read(teil);
            text1b = text1b + new String(teil,0,anzGeleseneBytes);
        }
        inBufStream.close();
        System.out.println("2. \n" + text1b);

// -----

```

// 2.a) Schreiben von Text und Zahlen

```
DataOutputStream outDataStream =
new DataOutputStream(
    new BufferedOutputStream(
        // 1) in Datei:
        new FileOutputStream("mytext2a.txt")
        // 2) auf Standardausgabe (i.d.R. Bildschirm bzw.
        // console)
        // System.out
    )
);
```

```
outDataStream.writeBytes("Die ist ein Text\nmit");
outDataStream.writeInt(2);
    // wird binär codiert und nicht als ASCII
outDataStream.writeBytes(" Zeilen.");
outDataStream.close();*/
```

// -----

// 2.b) Byteweises Schreiben in Datei

```
BufferedOutputStream outBufStream =
    new BufferedOutputStream(
        new FileOutputStream("mytext2b.txt"));
```

```
String text2b =
    new String("Die ist ein Text\nmit 2 Zeilen.");
// String byteweise speichern
outBufStream.write(text2b.getBytes()); // 8 bit ASCII
outBufStream.flush(); // erzwingen, dass Cache geleert wird
outBufStream.close(); // Datei schliessen
```

```
}
}
```