

— Übungsblatt 6 (T) —

## Aufgabe 19 (T)

(Gepufferte Ein- und Ausgabe)

Vorgegeben ist eine Text-Datei mit dem Namen `Eingabe.txt`, die zwei Zeilen Text enthält.

Schreiben Sie ein Programm, das zwischen diese beiden Zeilen die neue Textzeile `NEUE ZEILE` einfügt und die drei so gruppierten Zeilen in eine Datei mit dem Namen `Ausgabe.txt` speichert. Danach soll der Inhalt der Datei zur Kontrolle auch auf den Bildschirm ausgegeben werden.

Verwenden Sie jeweils geeignete gepufferte Streams.

## Aufgabe 20 (T)

(StreamTokenizer)

Die Klasse `java.io.StreamTokenizer` zur Zerlegung eines Input-Streams in Tokens stellt bekanntlich unter anderem folgende Komponenten bereit:

- `public static final int TT_EOF`  
Konstante die anzeigt, dass das aktuelle Token das *Stromende* ist.
- `public static final int TT_NUMBER`  
Konstante die anzeigt, dass das aktuelle Token eine *Zahl* ist.
- `public double nval`  
Wenn das aktuelle Token eine *Zahl* ist, enthält diese Komponente den Wert der Zahl.
- `public StreamTokenizer(Reader r)`  
Erzeugt ein `StreamTokenizer`-Objekt zum angegebenen Zeichen-Strom `r`.
- `public int nextToken() throws IOException`  
Liest das nächste Token aus dem Eingabe-Strom des `StreamTokenizers` und liefert den Typ des Tokens (z. B. *Stromende* oder *Zahl*) zurück.

Sie sollen nun mit Hilfe der `StreamTokenizer`-Klasse eine Klasse `NumberTester` realisieren, die beim Aufruf ihrer Methode `main` eine Text-Datei (ihr Name wird dem Programm `NumberTester` als Kommandozeilen-Argument übergeben) daraufhin untersucht, ob sie eine Zahl enthält, und entsprechende Information ausgibt.

Zunächst soll mit Hilfe des beim Aufruf angegebenen Dateinamens ein `File`-Objekt und ein `StreamTokenizer`-Objekt erzeugt werden. Danach sollen (mit Hilfe einer Schleife) die einzelnen Tokens, die das `StreamTokenizer`-Objekt liefert, untersucht werden. Falls ein Token eine Zahl ist, soll sofort die Meldung

Vorsicht! Der Text enthält die Zahl ...

auf den Bildschirm ausgegeben (die Pünktchen ... sind dabei durch den Wert der Zahl zu ersetzen) und die `main`-Methode beendet werden.

Die Schleife soll solange durchlaufen werden, bis das *Stromende* (Dateiende) erreicht ist.

Nach dem Ende der Schleife soll der Text

Der Text ist 'sauber'!

ausgegeben werden. Sie dürfen davon ausgehen, dass der Aufruf von `NumberTester` stets **mit** einem Kommandozeilen-Argument (dem Dateinamen) erfolgt.

**Aufgabe 21 (T)**
**(Wahr/Falsch)**

Sind die folgenden Aussagen wahr oder falsch? Bitte ankreuzen (pro Zeile höchstens ein Kreuz).  
 Für jedes korrekte Kreuz gibt es einen Punkt, für jedes inkorrekte Kreuz gibt es eine halben Punkt Abzug. Die minimale Gesamtpunktzahl ist aber 0.

Aussage	wahr	falsch
Ein Objekt der Klasse <code>BufferedReader</code> stellt einen Byte-Strom dar.		
Die Methode <code>read</code> der Klasse <code>InputStream</code> liefert ein Ergebnis vom Typ <code>byte</code> .		
Alle in einem Interface definierten Methoden ohne Rumpf sind implizit abstrakt.		
Die Methode <code>sleep</code> aus der Klasse <code>Thread</code> ist eine Instanzmethode.		
Der Name eines Threads wird bei seiner Erzeugung festgelegt und kann danach nicht mehr verändert werden.		
Ist <code>f</code> ein Objekt der Klasse <code>SimpleDateFormat</code> und <code>d</code> ein Objekt der Klasse <code>Date</code> , so liefert <code>d.format(f)</code> ein Ergebnis vom Typ <code>String</code> .		
Bei einem <code>PrintWriter</code> -Objekt wird das automatische Leeren des Ausgabe-puffers (flushing) durch die Methode <code>println</code> , aber nicht durch die Methode <code>print</code> ausgelöst.		
Die Methoden <code>wait</code> und <code>notify</code> sind Methoden, die in der Klasse <code>Thread</code> deklariert sind.		