

## Übung 6 [6+12+6]: Reader und Writer

Schreiben Sie einen FilterReader einen FilterWriter sowie ein Testprogramm nach folgender Spezifikation:

### CanonicalInputReader:

Die Zeichen werden einzeln eingelesen, wobei doppelte Leerzeichen entfernt werden, d. h. es gibt nie zwei aufeinanderfolgende Leerzeichen. Im Konstruktor ist ein Parameter zu übergeben, der festlegt, ob alle Kleinbuchstaben in Großbuchstaben umgewandelt werden sollen.

Zusätzlich sind Umlaute in entsprechende String zu konvertieren: z. B. "ä" wird zu "&uuml;". Behandelt werden sollen äöüßÄÖÜ. Informationen über die zur Ersetzung notwendigen Ergebniscodes finden Sie unter dem URL <http://www.uni-passau.de/~ramschi/iso8859-1.html>. Bei dieser Umwandlung müssen Sie trotz mehrmaliger (ev. einzelner) Leseanforderungen interne Daten weitergeben ohne vom zugrundeliegenden Stream einzulesen.

Alle Lesemethoden sind zu überschreiben. Mark, reset und skip sind nicht zu implementieren.

### FormatOutputWriter:

Es werden das Zeichen '\r' und Leerzeilen bei der Ausgabe unterdrückt (Zeilen mit einem einzelnen Leerzeichen sind NICHT leer!). Am Ende der Ausgabe (d. h. in close) wird eine Statistik ausgegeben: Anzahl der Zeilen und Anzahl der Zeichen. Am Beginn jeder Zeile ist eine Zeilenummer und die Anzahl der Zeichen in dieser Zeile auszugeben (Siehe Beispiel). Dazu müssen sie solange Zeichen in einem internen Buffer speichern, bis eine Zeile voll oder der Stream zu Ende ist.

Alle Zeichen einer Zeile nach "/" (inklusive der Schrägstriche) werden nicht ausgegeben. Weiters werden alle Zeichen zwischen "/\*" und "\*/" (inklusive dieser Zeichenketten) entfernt. Um das Ende eines Kommentars nicht durch einen innen enthaltenen Zeilenende-Kommentar (//) zu verlieren, sind zuerst die normalen Kommentare zu entfernen und erst anschließend die Zeilenende-Kommentare. Achtung: Ein Zeilenumbruch innerhalb eines Kommentars (z. B. "a/\*\n\*/b\n") wird entfernt (Ergebnis: "ab\n" = 1 Zeile und nicht "a\nb\n" = 2 Zeilen)!

## Testprogramm:

Schreiben Sie ein Programm zum Testen der beiden Klassen. Folgende Parameter werden akzeptiert:

- `-nocase`: Alle eingelesenen Zeichen sollen vom `CanonicalInputReader` in Großbuchstaben umgewandelt werden.
- `-i Inputfile`: Wenn `-i` als Option angegeben wird, so ist der nächste Parameter der Name einer Datei, von der die Eingabe eingelesen wird. Fehlt dieser Parameter wird von der Tastatur eingelesen.
- `-o Outputfile`: Wenn `-o` als Option angegeben wird, so ist der nächste Parameter der Name einer Datei, auf welche die Ausgabe geschrieben wird. Fehlt dieser Parameter wird am Bildschirm ausgegeben.

Die Parameter sind zu überprüfen (Existenz, Lesbarkeit, ...).

Die einzige Tätigkeit des Programms ist, die Parameter einzulesen, ev. nötige Dateien zu öffnen, die Streams anzulegen und anschließend vom Eingabestream (Reader) solange zum Ausgabestream zu kopieren, wie Daten vorhanden sind.

## Beispiel 1:

```
1 - 7: Zeile 1
2 - 22: Zeile 2 - etwas länger
3 - 7: Zeile 3
Insgesamt 3 Zeilen mit 36 Zeichen
```

## Beispiel 2:

```
out.write("Zeile /*1//comment*\nZei*/le */** 2**/*\n");
out.write("\n\n// ghsdfgh sdf hsdf\n");
out.write("Zeile /*3\nZeike */4//asdkfhasdfa\n\n");
```

```
1 - 6: Zeile
2 - 10: Zei*/le **
3 - 7: Zeile 4
Insgesamt 3 Zeilen mit 23 Zeichen
```