

## 7. Übung

**Abgabe: bis 20.12.2022, 6 Uhr**

**Abnahme DFHI: 20.12.2022**

**Abnahme PI: 21.12.2022**

### Aufgabe 1 - Link-Filter

Schreiben Sie ein Java-Programm, das aus HTML-Dateien die Hypertextlinks herausfiltert und ausgibt. Da wir die Dateiverarbeitung noch nicht besprochen haben, lesen Sie die Datei am einfachsten mit dem `cat`-Kommando ein, das Sie in der Vorlesung Betriebssystemeinführung gelernt haben. Programmieren Sie dazu eine Texteingabe mit dem Scanner, d.h. der Benutzer wird aufgefordert, den HTML-Text einzugeben (nicht den Pfad der Datei) und dann liest der Scanner den html-Text über die Standardeingabe ein. So können Sie auch html-Code zum Testen eingeben. Starten Sie Ihr Programm dann entweder mit dem Kommando `cat datei | java IhrJavaProgramm` oder mit `java IhrJavaProgramm < datei`. Auf diese Art wird der Inhalt der Datei mit dem Scanner eingelesen.

Neben den Links selbst sollte Ihr Programm noch die folgenden Zusatzinformationen ausgeben:

- Anzahl gefundener Links und Anzahl der Zeilen
- Anzahl der Zeichen in der URL

Die Hypertextlinks werden durch das `a`-Element definiert (siehe dazu auch <https://wiki.selfhtml.org/wiki/HTML/Textauszeichnung/a>).

Die Datei `test1.html` sieht bspw. folgendermaßen aus:

```
<p>
<li><a href="http://www.htwsaar.de/">htw saar</a></li>
<li><a href="http://www.google.com">Google</a></li>
<a href="http://www.heise.de">Heise Online</a>
</p>
```

Die Ausgabe sollte in diesem Fall **ungefähr** so aussehen:

```
htw saar:      http://www.htwsaar.de/, Anzahl Zeichen: 22
Google:        http://www.google.com, Anzahl Zeichen: 21
Heise Online:  http://www.heise.de, Anzahl Zeichen: 19
3 Links wurden in 5 Zeilen gefunden.
```

Sie können zum Filtern der Dateien die Methoden der Klassen `String` verwenden oder reguläre Ausdrücke. Gehen Sie der Einfachheit halber von folgenden Annahmen aus:

- die Hypertextlinks stehen immer vollständig in einer Zeile
- die Texte in den Tags sind immer kleingeschrieben

Erstellen Sie eine einfache eigene Testdatei, die genügend Daten enthält, um alle oben erwähnten Annahmen zu überprüfen. Ein JUnit-Test für diese Aufgabe ist möglich, aber schwierig umzusetzen. Aus diesem Grund genügt eine einfache HTML-Testdatei, die alle relevanten Fälle beschreibt. Das Ergebnis müssen Sie dann von Hand auswerten. Achten Sie auf guten objektorientierten Programmierstil. Verwenden Sie für die Ausgabe die String-Formatierung.

Wenn Sie mehr über HTML erfahren wollen, ist die folgende Seite sehr zu empfehlen:

<https://selfhtml.org>.