

## Übungsblatt 9 zur Vorlesung Programmieren (fällig bis 25.01.26)

### Aufgabe 1 (Exceptions, java.io):

In dieser Aufgabe soll ein Programm erstellt werden, das eine Textdatei einliest und die Anzahl der Zeilen, Wörter und Zeichen der Datei bestimmt.

Dazu sollen die Klassen `FileReader` und `BufferedReader` aus dem `java.io` Paket verwendet werden. Zum Zerlegen einer Zeile in Worte können Sie die `split(...)`-Methode benutzen (in der Dokumentation der Klasse `String` finden Sie einen Verweis auf mögliche Regex-Patterns für die `Split`-Funktion). Whitespace-Zeichen am Anfang und Ende einer Zeile können Sie im Voraus mit einer geeigneten Methode der `String`-Klasse entfernen.

- a) Lesen Sie sich die offizielle (!) Dokumentation zu den oben genannten Klassen sowie zur Klasse `StringBuilder` durch.
- b) Erstellen Sie eine Klasse `CharacterCounter` und implementieren darin die Methoden

```
String readFileContent(String filename) und  
Counter count(String content)
```

Die erste Methode soll den gesamten Inhalt einer Datei einlesen und als `String`-Wert zurückgeben. Das Einlesen soll zeilenweise und gepuffert (mittels eines `BufferedReaders`) erfolgen. Der Rückgabewert soll mittels eines `StringBuilder`s erstellt werden.

Die zweite Methode soll das Zählen von Zeilen, Wörtern und Zeichen bewerkstelligen. Der Rückgabewert soll ein Objekt der Klasse `Counter` sein, das drei Attribute für die drei Zählwerte besitzt.

- c) Fügen Sie der Klasse `CharacterCounter` eine `main`-Methode hinzu. Das erste Argument des `String`-Arrays soll dabei den Dateinamen enthalten.
- d) Welche Exceptions können von den von Ihnen verwendeten `java.io`-Methoden geworfen werden? Stellen Sie sicher, dass sämtliche Exceptions dieser Methoden in Ihrer `main`-Methode abgefangen und sinnvoll behandelt werden. Die Methoden `readFileContent` und `count` sollen die Exceptions selbst nicht behandeln, sondern diese an die aufrufende Methode (`main`) weitergeben.

### Aufgabe 2 (Unittests mit JUnit):

- Überlegen Sie sich Unit-Tests für die Methode `count` Ihres Programms aus Aufgabe 1. Begründen Sie, warum Sie diese Tests für sinnvoll und ausreichend erachten.
- Erstellen Sie mindestens drei Unit-Tests und implementieren Sie diese mittels JUnit (siehe Foliensatz P7). Je nach verwendeter Java-Version kann es nötig sein, JUnit zum Klassen- oder zum Modulpfad hinzuzufügen. Ihre Entwicklungsumgebung unterstützt Sie dabei durch entsprechende Vorschläge.