

13. Übung zu Programmierung 1**Abgabe: freiwillig**

Die folgende Aufgabe war bereits in der 11. Übung gestellt worden. Die Funktionalität ist nun gemäß der 2. Aufgabe mit Hilfe von Collection-Klassen zu erweitern.

1. Aufgabe

Erstellen Sie ein Programm mit der folgenden Funktionalität:

Aufruf : java LOCAuswertung datei1.java datei2.java datei3.java

Wirkung: Es werden die sogenannten „Lines of code (LOC)“ gezählt, d. h. die Anzahl an relevanten Codezeilen in den zu untersuchenden Java-Quelltexten.

Zu beachten ist dabei Folgendes:

- Es ist mindestens eine Datei zu übergeben. Die Gesamtanzahl an übergebenen Dateien ist beliebig.
- Die zu verarbeitenden Dateien sind auf die Eigenschaft "normale Datei" und "Lesbarkeit" zu prüfen.
- Mögliche Ausnahmen sind zu behandeln. Dazu ist eine eigene Ausnahmeklasse zu definieren.
- Bei Lesefehler in einer Datei soll mit der nächsten Datei fortgefahren werden.
- Zu zählen sind dabei alle nichtleeren Zeilen, die keine Kommentarzeilen sind.
 - Dabei können leere Zeilen durchaus eine Länge größer als 0 haben.
 - Als Kommentarzeilen betrachten wir der Einfachheit halber nur die Zeilen, die mit dem String "`/*`" beginnen.
- Die Ausgabe des Programmes sollte dabei etwa wie folgt aussehen:

Aufruf:

```
java LOCAuswertung Ausdruck.java Summe.java
```

Ausgabe:

```
Auswertung Lines Of Code (LOC)
Ausdruck.java: 56 LOC
Summe.java: 23 LOC
```

```
Gesamt:
2 Dateien 79 LOC
```

2. Aufgabe

In dieser Aufgabe wird die Funktionalität der ersten Aufgabe um die folgenden Dinge erweitert:

- Es werden zusätzlich auch noch die Kommentarzeilen gezählt, also die Zeilen, die nur aus Kommentar bestehen.
- Die ermittelten Daten über Dateinamen, LOC und Kommentarzeilen werden in eine geeignete Datenstruktur übertragen. Dafür sollten natürlich Collections verwendet werden.
- Die Daten sollen nun auf die folgenden Arten ausgewertet und ausgegeben werden:
 - Ausgabe sortiert nach Dateinamen mit den Informationen LOC, Kommentarzeilen und Comment Ratio (Verhältnis zwischen Anzahl Kommentarzeilen zu LOC)
 - Ausgabe absteigend sortiert nach LOC
 - Ausgabe absteigend sortiert nach Comment Ratio