

Praktikum 6 zu Objektorientierte Programmierung

Aufgabe 6.1 (Aufwand ca. 5 Stunden, Abgabe bis 25.05.2014 um 18 Uhr)

Viele byte-orientierte Datenströme enthalten Text „in Klarschrift“. Zum Beispiel können digitale Fotos die Angabe des Kamerateyps und andere Aufnahmeinformationen enthalten. Zip-Dateien enthalten die Namen der gepackten Dateien und übersetzte Java-Klassen (class-Dateien) enthalten die Namen von Klassen und Variablen im Klartext.

In dieser Aufgabe sollen Sie eine Klasse `Textfinder` schreiben, die „lesbare“ Zeichenfolgen aus einem byte-orientierten Datenstrom herausfiltert. Betrachten Sie dafür einen Bytewert b als dasjenige Zeichen, dessen Code b ist. Beispiel: der Bytewert 65 steht für das Zeichen `a`, da dieses Zeichen den Code 65 besitzt.

Eine lesbare Zeichenfolge ist eine Bytefolge, die mit einem Buchstaben (`a-z` und `A-Z`, keine Umlaute) beginnt, gefolgt von beliebig vielen Buchstaben oder Ziffern. Der Textfinder soll die längstmöglichen „lesbaren“ Zeichenfolgen finden.

Realisieren Sie im Paket `bytefolge` eine Klasse `Textfinder` mit folgenden Methoden:

- Einen Konstruktor `Textfinder(InputStream, int)`, der einen Textfinder für die übergebene Datenquelle erzeugt. Der Textfinder soll nur Wörter finden, die mindestens die angegebene Länge besitzen.
- Eine Methode `Set gibWoerter()`, die die Menge aller Wörter, die der Textfinder in der Datenquelle gefunden hat, zurückgibt.
- Eine Methode `int gibHaeufigkeit(String)`, die angibt, wie häufig das übergebene Wort in dem Datenstrom gefunden wurde. Wurde das Wort nicht gefunden, soll 0 zurückgegeben werden.

Da der Eingabestrom eine unvorhersehbare Länge haben kann, dürfen Sie die Datenquelle nicht zunächst komplett einlesen, sondern müssen die Worterkennung während des Einlesens durchführen.

Realisieren Sie im Paket `bytefolge` außerdem eine Testklasse basierend auf JUnit, um das Verhalten der Klasse `Textfinder` und ihrer Methoden zu testen. Erinnern Sie sich an die Vorlesung, in der wir besprochen haben, welcher Eingabestrom zum Testen geeignet ist.

Hinweise

- Achten Sie darauf, dass in den Eigenschaften Ihres NetBeans-Projekts die Zeichenkodierung UTF-8 eingestellt ist. Ist dies nicht der Fall, kann es bei der automatischen Auswertung Ihrer Lösung zu Fehlern beim Compilieren kommen. Die Lösung wird dann mit einer Erfolgsquote von 0% gewertet.
- Verwenden Sie nur den Vorlesungsstoff bis einschließlich Kapitel 9.
- Sie dürfen selbstverständlich zusätzliche Methoden realisieren.
- Denken Sie an die ausreichende Dokumentation und Kommentierung Ihrer Lösung. Beachten Sie die unterschiedliche Bedeutung der *externen Dokumentation* `/** ... */` vor einer Klasse oder Methode und des *Implementierungskommentars* `/* ... */` innerhalb einer Methode. Die externe Dokumentation sagt, *was* eine Klasse oder Methode leistet, der Implementierungskommentar hilft zu verstehen, *wie* es gemacht wird. Verwenden Sie Implementierungskommentare vor allem, um den Berechnungsablauf verständlich zu machen.
- Erzeugen Sie die HTML-Dokumentation Ihrer Klasse und überzeugen Sie sich, ob Ihre externe Dokumentation sinnvoll und ohne Kenntnis des Quellcodes der Klasse hilfreich ist.
- Erstellen Sie je ein zip-Archiv des Quellordners `bytefolge` unter `src` und des gleichnamigen Ordners unter `test` und laden Sie beide zu Moodle hoch.
- Im Veranstaltungskalender finden Sie die Termine, an denen diese Lösung im Praktikum besprochen wird. Bringen Sie zu diesen Terminen bitte die Auswertung Ihrer Lösung mit, entweder ausgedruckt oder unmittelbar auf Ihrem Rechner verfügbar.