

# Algorithmen und Datenstrukturen

## Praktikum

SoSe 22

Prof. R. Zahoransky (Aufgaben von Herrn Prof. Rülling)

Hochschule Furtwangen (University of Applied Science)

### Aufgabenblatt 9

In dieser Übung ist ein Java-Programm vorgegeben, das die Wörter einer Textdatei einliest und am Monitor anzeigt.

#### Aufgabe 1

Ergänzen und modifizieren Sie das Programm, so dass die Wörter nicht mehr angezeigt, sondern in ein Wörterbuch eingetragen werden. Verwenden Sie als Wörterbuch die Java-Implementierung `Tree-Map`. Ein Beispiel zur Nutzung:

```
import java.util.Map;
import java.util.TreeMap;

public class TreeMapTest {

    public static void main(String[] args) {
        Map<Integer, String> map = new TreeMap<>();
        map.put(42, "Datenstrukturen");
        map.put(8, "und");
        map.put(5, "Algorithmen");

        for (String eintrag: map.values()) {
            System.out.println(eintrag);
        }
    }
}
```

Die Ausgabe ist an Hand der Schlüssel (42, 8, 5) aufsteigend sortiert und lautete „Algorithmen und Datenstrukturen“.

Am Ende sollen die Wörter alphabetisch sortiert mit der Anzahl ihres Vorkommens im Text ausgegeben werden. D.h., so wie in Übung 3, in der gezählt wurde, wie oft bestimmte Buchstaben in einem Text vorkamen, soll jetzt gezählt werden, wie oft einzelne Wörter im Text vorkamen. **Tipp:** Vielleicht ist es hilfreich zwei Wörterbücher zu benutzen?

#### Aufgabe 2

Erweitern Sie Ihr Programm um einen Zeilenzähler, so dass beim Lesen eines Wortes auch bekannt ist, in welcher Zeile es steht. Benutzen Sie diese Information schließlich, um am Ende für jedes Wort die Liste der Zeilennummern seines Vorkommens auszugeben.

#### Aufgabe 3

Wenden Sie schließlich Ihr Programm auf seine eigene Java-Quelldatei an, um seine Referenz-Liste für die verwendeten Bezeichner und Schlüsselwörter zu erhalten.

```

public class AnalyzeText {
    private static boolean isLetter(char c) {
        return (((('a' <= c) && (c <= 'z')) || (('A' <= c) && (c <= 'Z')) ||
                (c == 'ß') || (c == 'ä') || (c == 'ö') || (c == 'ü') ||
                (c == 'Ä') || (c == 'Ö') || (c == 'Ü')));
    }

    private static String readWord(BufferedReader f) throws IOException {
        String s = new String();
        char c;
        int inputChar;
        // Auf nächstes Wort warten
        do {
            inputChar = f.read();
            c = (char) inputChar;
        } while (!isLetter(c) && (inputChar != -1));
        // Auf Wortende warten
        while (isLetter(c)) {
            s += c;
            inputChar = f.read();
            c = (char) inputChar;
        }
        if (s.isEmpty())
            return null;
        else
            return s;
    }

    public static void main(String[] args) {
        try {
            BufferedReader f = new BufferedReader(new
FileReader("/Pfad/zur/Datei"));
            String s;
            while ((s = readWord(f)) != null) {
                System.out.println("Gelesenes Wort:" + s);
            }
        } catch (IOException e) {
            System.out.println("Fehler beim Lesen:" + e.getMessage());
        }
    }
}

```