

Übungsblatt 9

Abgabe bis Dienstag, den **3. Juli** um **12:00** Uhr

Bei diesem Übungsblatt geht es darum die *k-häufigsten* Zeichenketten in gegebenen Dateien zu finden. Ein Zeichenkette muss dabei eine Länge von mindestens 10 Zeichen haben und besteht aus den Zeichen *a – z*. Allgemeine Tests werden wieder zur Verfügung gestellt, erweitern Sie diese um Spezialfälle und Anforderungen aus den Aufgaben.

Aufgabe 1 (5 Punkte)

Deklarieren und implementieren Sie eine Methode *updateWordCounts* einer Klasse *WordCounter*, welche eine Liste von Zeichenketten entgegennimmt. Wenn eine Zeichenkette mindestens die in *_minLength* gespeicherte Länge hat, so soll der Zähler dieser Zeichenkette je Vorkommen in der Argumentliste um 1 erhöht werden. Der Member *_minLength* soll nur positive Ganzzahlen speichern und standardmäßig den Wert 10 haben.

Deklarieren und implementieren Sie eine Methode *getWordCount* der Klasse *WordCounter*, welche die Anzahl Vorkommen einer Zeichenkette zurückgibt. Sollte eine Zeichenkette zuvor nicht gespeichert gewesen sein, so darf diese danach ebenfalls nicht gespeichert sein.

Aufgabe 2 (5 Punkte)

Deklarieren und implementieren Sie eine Methode *readFile* der Klasse *WordCounter*, welche eine gegebene Datei zeilenweise einliest, Großbuchstaben in Kleinbuchstaben umgewandelt, in maximale Zeichenketten der Elemente *a – z* aufteilt und dann die Anzahl Vorkommen aktualisiert.

Deklarieren und implementieren Sie eine Methode *getNumWords* der Klasse *WordCounter*, welche die Anzahl der gespeicherten Zeichenketten zurückgibt.

Aufgabe 3 (5 Punkte)

Deklarieren und implementieren Sie eine Methode *computeFrequentWords* der Klasse *WordCounter*, welche die *k-häufigsten* Zeichenketten berechnet und diese mit ihren Häufigkeiten zurückgibt. Dabei sollen die Zeichenketten nach ihrer Häufigkeit absteigend sortiert sein, der Wert für *k* soll im Member *_topK* als positive Ganzzahl gespeichert sein und standardmäßig den Wert 15 haben.

```
[#include <algorithm>; std::please_turn_over();]
```

Aufgabe 4 (5 Punkte)

Schreiben Sie ein ausführbares Programm (in *WordCounterMain.cpp*), welches die 15 häufigsten Zeichenketten einer Datei (gegeben in *argv[1]*) mit ihren Häufigkeiten ausgibt. Sollten weniger Zeichenketten die Bedingungen erfüllen, sollen entsprechend weniger Ergebnisse ausgegeben werden.

Führen Sie das Programm auf jeder der auf dem Wiki verlinkten Dateien aus und schreiben sie die (bis zu) 15 häufigsten Zeichenketten pro Datei in Ihre *erfahrungen.txt*.

Aufgabe 5 (Optional)

Erweitern Sie Ihr Programm um den Kommandozeilenparameter *--min-length/-l*, um die minimale Länge zu verändern (*_minLength*, siehe Aufgabe 1), sowie um *--top-k/-k*, um die Anzahl der Ergebnisse zu verändern (*_topK*, siehe Aufgabe 3). Teilen Sie die Behandlung der Argumente, wie in Vorlesung 7 gelernt, in Methoden *processCommandLineArguments* und *process* der Klasse *WordCounter* auf.

Laden Sie wie gewohnt alle Code-Dateien und das Makefile in unser SVN hoch, in einem neuen Unterverzeichnis *blatt-09*. Es gelten weiterhin die 10 Gebote von der letzten Seite des Ü1, speziell Gebot 6 ist zu beachten. *valgrind*, *Zugriffsmodifikatoren*, *const-correctness*, *et cetera* werden an dieser Stelle nicht mehr explizit erwähnt.

Laden Sie wie gehabt auch eine Datei *erfahrungen.txt* hoch (im Unterordner *blatt-09*), in der sie kurz Ihre Erfahrungen mit dem Ü9 und der Vorlesung dazu beschreiben.

Kann eine Mannschaft bei der WM die Vorrunde überstehen und ins Achtelfinale einziehen, ohne ein einziges Tor zu schießen?