

Übungsblatt 9

Abgabe bis Mittwoch, den 30. Juni um 16 Uhr

Aufgabe 1 (10 Punkte)

Ein Anagramm ist ein Wort, das durch Umstellung der Buchstaben aus einem anderen Wort gebildet wurde. Zum Beispiel ist *Gehirn* ein Anagramm von *Hering*.

Ziel dieser Aufgabe ist es, ein Programm *AnagramFinderMain* zu schreiben, das für ein gegebenes Wort alle dessen Anagramme in einem vorgegebenen Wörterbuch findet. Benutzen Sie dabei das Wörterbuch im SVN in *vorlesungen/vorlesung-9/dictionary.txt*; es enthält ein Wort pro Zeile und insgesamt 311,177 Wörter. Es soll nicht zwischen Groß- und Kleinschreibung unterschieden werden. Für das Wort *Ostern* sollten Sie zum Beispiel die folgenden 6 Anagramme finden: *Ostern*, *Ortens*, *Sorten*, *Tenors*, *Tensor*, *rosten*.

Wie gehabt sollen Sie das Programm natürlich mit einer sinnvollen Klassen realisieren und in der *AnagramFinderMain.cpp* sollten nur wenige Zeilen Code stehen, die von dieser Klasse Gebrauch machen. Einen Vorschlag für das Design so einer Klasse finden Sie in *AnagramFinder.h*; dabei werden die in der Vorlesung vorgestellten STL-Klassen *vector* und *string* verwendet. Sie müssen sich aber nicht an diese Vorlage halten und können die Klasse schreiben wie Sie möchten, solange es sinnvoll ist und am Ende alles funktioniert.

Wie immer sollen Sie für jede Methode einen guten Test schreiben, anders gibt es keine Sicherheit, dass Ihr Programm das tut, was es tun soll.

Wer eines der Worte mit den meisten Anagrammen in dem genannten Wörterbuch findet, bekommt einen Bonuspunkt.

Committen Sie wie gehabt alles ins SVN, in ein neues Unterverzeichnis *uebungsblatt-9*, stellen Sie sicher, dass alles kompiliert, die Tests ohne Fehler laufen, Lint ohne Fehler durchläuft und auch auf Hudson alles funktioniert. Und die *erfahrungen.txt* nicht vergessen.