

## Agile Webentwicklung mit Ruby on Rails

*Sommersemester 2012*

### Blatt 1

#### Aufgabe 1: Sortieren

Übersetzen Sie einen oder mehrere aus der Veranstaltung *Algorithmen und Datenstrukturen* bekannten Sortieralgorithmus Ihrer Wahl von Java nach Ruby.

Schreiben Sie dazu eine Methode, die nach dem Sortieralgorithmus benannt ist und als Parameter ein Array mit ganzen Zahlen erwartet. Der Rückgabewert soll das sortierte Array sein.

#### Aufgabe 2: Array summieren

Es sei ein Array mit ganzen Zahlen gegeben. Finden Sie mindestens drei verschiedene Möglichkeiten dessen Elemente aufzusummieren.

*Hinweis: Die API-Dokumentation der Klasse Array könnte hilfreich sein.*

#### Aufgabe 3: Romeo und Julia

Ergänzen Sie die beiliegende Datei `text_analysis.rb` um eine Methode `rare_words(word_list, number)`, welche die `number` am seltensten vorkommenden Wörter aus der übergebenen Wortliste heraussucht und einen

Hash zurückliefert, dessen Schlüssel die Wörter und dessen Werte die Anzahl sind.

Betrachten Sie die beiliegende Datei `romeo_and_juliet.txt`, sie enthält den Volltext des Stückes *Romeo und Julia* in englischer Sprache.

Erstellen Sie eine Datei namens `romeo_and_juliet_analysis.rb`, welche die folgenden Analysen durchführt:

- Ausgabe einer Liste der `n` seltensten Wörter, `n` soll als Parameter auf der Kommandozeile übergeben werden können. Wird kein Parameter übergeben, so sollen die fünf seltensten Wörter ausgegeben werden.
- Ausgabe der Textzeilen, welche die seltensten Wörter enthalten. Beachten Sie dabei die Groß- und Kleinschreibung, die erzeugten Wortlisten enthalten nur klein geschriebene Wörter, im Ausgangstext sind diese aber nicht unbedingt klein geschrieben.
- Ausgabe des Redeanteils der auftauchenden Charaktere. Damit ist nicht die Anzahl der gesprochenen Wörter gemeint, sondern lediglich, wie oft die Person redet.

Beachten Sie außerdem, dass vor und nach dem eigentlichen Text von *Romeo und Julia* noch Lizenzhinweise stehen, die **nicht** verarbeitet werden sollen.

Lagern Sie diese Analysen nach eigenem Ermessen aus Gründen der Übersicht in weitere Dateien und Methoden aus.

*Hinweis: Regular Expressions und die API-Dokumentation der Klasse File könnten hilfreich sein.*

## Aufgabe 4: Array#sample!

Erweitern Sie die Core-Klasse `Array` um eine Methode `sample!`, welche ein zufälliges Element aus dem Array zurückliefert und dieses aus dem Array entfernt. Beachten Sie dabei, dass bei mehrfach vorkommenden Elementen nur **eins** davon entfernt werden darf.