

Agile Webentwicklung mit Ruby on Rails

Sommersemester 2012

Blatt 3

Aufgabe 1: Vector Refactoring

Betrachten Sie die vom letzten Übungsblatt bekannte (und hier beiliegende) Datei `vector.rb`. Führen Sie ein geeignetes Refactoring der Klasse durch.

Aufgabe 2: City

Die Klasse `City` repräsentiere eine Stadt in einem Spiel. In einer Stadt können verschiedene Ressourcen gelagert werden, die erst im Laufe des Spiels bekannt werden.

Schreiben Sie ein *Class Macro* `add_resource(name)`, welches für die Ressource `name` lesenden Zugriff zur Verfügung stellt. Zur Manipulation der Ressourcen muss jeweils eine Methode zum Inkrementieren und zum Dekrementieren definiert werden. Die Ressource `iron` wäre also lesbar über die Methode `iron`, kann mit `increment_iron!` um eins erhöht und mittels `decrement_iron!` um eins verringert werden.

Um Ressourcen dynamisch hinzufügen zu können, bekommt der Initializer von `City` einen *Hash Parameter*, dessen Schlüssel die Namen der Ressourcen und deren Werte die Anzahl der initialen Einheiten sind. Verwenden Sie das vorher erstellte *Class Macro*, um entsprechende Methoden zur Interaktion mit den Ressourcen zu definieren. Beachten Sie dabei, dass das *Class Macro* nur aufgerufen werden darf, wenn die Ressource noch nicht bekannt ist.

Aufgabe 3: AppConfig

Zur Konfiguration einer Applikation muss eine Klasse `AppConfig` erstellt werden. Entsprechende Objekte sollen beliebige Einstellung verwalten können. Ein Aufruf der Art `conf.some_value = 42` soll zur Folge haben, dass dieser Wert danach mittels `conf.some_value` wieder lesbar ist. Weiterhin soll über `conf.some_value?` ermittelbar sein, ob die Einstellung `some_value` gesetzt worden ist. Verwenden Sie *Ghost Methods* zur Umsetzung.