



Praktikum zu LV 7328 - Ruby: **Übung 12 (Finale!)**

Software-Entwicklung mit Ruby:
Dokumentation mit RDoc
Unit Tests mit Test::Unit



Organisatorisches



- Arbeitsverzeichnis:
`~/lv/ruby/12/`
- Dateinamen:
`12-lotto.rb` # Entwickeln und abgeben
`12-lotto.tar.gz` # Verzeichnis incl. RDoc
`sets.rb` # Beispiel-Code für Unit testing
- Werkzeuge:
`ruby` # Der Interpreter
`emacs` # mit Ruby-Mode. Auch X-Emacs ok
`rdoc, irb, ri` # Neu: rdoc (!)
- Abzugeben:
`12-lotto.rb`
`12-lotto.tar.gz`



A: Unit testing



- Teilschritte
 - Legen Sie Datei 12-lotto.rb durch Kopieren von 03-lotto.rb an
 - Entfernen Sie die Kommandozeilen-Anteile für den Dialog, so dass nur noch die Implementierung der Klasse Lotto übrig bleibt.
 - Ergänzen Sie nun Test-Code mittels Modul Test::Unit
- Hinweise & Vorgaben
 - Erzeugen Sie eine Testklasse „TestSet“ und darin für jede zu testende Methode eine eigene Test-Methode gleichen Namens!
 - Für konkrete Tests benötigen Sie bestimmte Eingabewerte. Wählen Sie (a) typische Werte (wie 6 und 49), (b) Werte im Grenzbereich des Zulässigen, (c) Werte knapp außerhalb der zulässigen Bereiche.
 - Nicht übertreiben: Je ein gut gewähltes Beispiel soll genügen...
- Testen Sie die Tests:
 - Verfälschen Sie die Implementierung einer Methode vorübergehend. Wird die Störung erkannt?



A: Unit testing



- **Tests für „initialize“ bzw. „new“:**
 - Lässt sich Lotto.new mit 0 und 1 Parameter aufrufen?
 - Erscheint ein Fehler, wenn mehr Parameter übergeben werden?
- **Tests für „draw_one“**
 - Erhalten Sie wirklich stets eine Zahl zwischen 1 und „max“ ?
 - Erhalten Sie jede Zahl wirklich nur einmal??
- **Tests für „draw_all“**
 - Erhalten Sie ein Array aus n Zahlen zwischen 1 und „max“?
 - Testen Sie auch die Fälle $n < 0$, $n = 0$ und $n > \text{max}$!
- **Tests für „draw“**
 - Wie bei „draw_all“, aber hier müssen Sie das Rückgabe-Array erst selbst bilden.
- **Tests für „draw_sorted“**
 - Wie bei „draw“. Zusätzlich: Sind die Zahlen wirklich sortiert?



A: Unit testing – ein Beispiel



- Nutzen Sie die Datei „sets.rb“ als Beispiel-Code
 - Anlegen einer Klasse TestSet durch Ableiten von Test::Unit::TestCase genügt.
- Eine kleine Einführung finden Sie hier:
<http://www.rubygarden.org/ruby?UsingTestUnit>
- Wer es hat: Pickaxe-Buch (2. Auflage!), Kap. 12



B: Dokumentation



- Teilschritte
 - Fügen Sie Dokumentation als Kommentare in den Quellcode ein
 - Wenden Sie „rdoc“ auf Ihre Datei an
 - Prüfen Sie den erzeugten HTML-Output auf Vollständigkeit, Verständlichkeit und korrektes Funktionieren Ihrer Formatierungen
- Hinweise & Vorgaben
 - Die Beschreibung für rdoc finden Sie hier:
<http://rdoc.sourceforge.net/doc/index.html>
 - Lesen Sie insbesondere Abschnitte „Example“ und „Markup“!
 - Beschreiben Sie die Klasse „Lotto“ mit einem Kommentarblock
 - Fügen Sie Kommentar für jede Methode von „Lotto“ ein, achten Sie darauf, dass Eingabe-Parameter und Rückgabewerte klar beschrieben sind.
 - Verwenden Sie auch etwas „Markup“ zur Formatierung
 - Verhindern Sie die Erzeugung von Dokumentation zu den Test-Methoden!



- Geben Sie zunächst 12-lotto.rb separat ab
 - Die Unit Tests sollten funktionieren
 - Kommentare für RDoc sollten erhalten sein
- Erzeugen Sie dann ein tar-Archiv 12-lotto.tar.gz, das Ihr Arbeitsverzeichnis zu dieser Aufgabe enthält.
 - Die von rdoc erzeugten Unterverzeichnisse und Dateien können Sie auf diese einfache Weise also ebenfalls abgeben,