

Ruby初級者向けレッスン第 32 回

okkez@Ruby関西

今回の内容

- Test::Unit の使い方
- テスト駆動開発

今回のゴール

- Test::Unit の使い方を知る
- テスト駆動開発を体験する

Ruby でのテスト

- テストとは？
- Test::Unitの基礎
- いろんなテストメソッド
- ランナー
- Rake で実行

テストとは？

- 起こるべき動作を記述する
- テストを見れば仕様がわかる
- コードの変更を安心して行える

Test::Unit

- 助田さんによる RubyUnit が始まり
- Ruby1.8 では test/unit として標準添付
- Ruby1.9 では test/unit とそこそこ互換性のある minitest が標準添付されている
 - test/unit は Gem としてより新しいバージョンがあります

Test::Unit の基礎

- 「 $1 + 1 = 2$ であるべき」のテスト
- 配布資料参照 (デモ)

Test::Unit の基礎

- テストクラスは Test::Unit::TestCase を継承する
- テストメソッドは 'test' で始める

Test::Unit の基礎

- 前準備と後処理
 - setup 各テストメソッドが呼ばれる前に呼ばれる
 - teardown 各テストメソッドが呼ばれた後に呼ばれる

Test::Unit の基礎

- `assert(boolean, msg = nil)`
- `assert_match(pattern, string, msg = "")`
- `assert_raise(expected_exception_klass, msg = ""){ ... }`
- 詳細はリファレンスマニュアルを参照

Test::Unit の小技

- --help が有効 (minunit ではできません)
- 複数のテストファイルの連続実行
- Rake を使ってテストを実行する
- テスト結果をカラフルにしよう

その他のテストフレームワーク

RSpec

- <http://rspec.info/>

expectations

- <http://expectations.rubyforge.org/>

minunit

- <http://rubyforge.org/projects/bfts>

XP (eXtreme Programming)

- 個人の責任と勇気を重んじる人間中心の開発手法
- コミュニケーション、シンプル、フィードバック、勇気、尊敬の五つの価値
- ペアプログラミング、テスト・ファースト、短期リリースなどのプラクティス

テスト駆動開発

- まずテストを書く -> 失敗
- 次にコードを書く -> 成功
- リファクタリング -> 成功

リファクタリングとは

- 仕様を保持したままコードを変更する
- だから、仕様が決まっていないとリファクタリングしようがない
- だから、テストがないと「仕様の保持」が保証できない

by Martin Fowler

「リファクタリング中に2、3日システムが動かなくなっちゃってーなどと言ってる奴がいたら、んなもんリファクタリングじゃあなーいと言ってやれ」

Fake it!

- テストがおとるだけの最低限の実装をすることを Fake it! という
- ちゃんとした実装に時間がかかりそうでも、まずは書くべきテストを短時間で書く気持ちになれる

Triangulation

- 三角測量のこと
- 二つ以上のテストを書くことで Fake it! ではテストが通らなくする
- これにより、正しい実装へと導く

演習

- スタックを作ろう
- スタックの実装に連結リストを使ってみよう
- 逆ポーランド記法電卓を作ろう

テスト駆動開発をすると...

- 全てのコードがテストを通った状態で完成する
- テストしやすい設計をするようになる
- よりよいモジュール分割をするようになる
- よりよいインタフェースを作るようになる
- 開発にリズムができて楽しくなる

テスト駆動開発の主なアプローチ

- テストを書く (失敗) -> フェイクする (成功) -> リファクタリング (成功)
- テストを書く (失敗) -> フェイクする (成功) -> テストを足す (失敗) -> リファクタリング (成功)
- テストを書く (失敗) -> 明白実装 (成功)

まとめ

- テスト駆動であなとも上品プログラマ
- いろんなテストメソッドを使いこなそう
- `testrb` は意外と便利

今後の情報源

公式Webサイト

- <http://www.ruby-lang.org/>

リファレンスマニュアル

- <http://www.ruby-lang.org/ja/man/>

日本Rubyの会

- <http://jp.rubyist.net/>

Rubyist Magazine

- <http://jp.rubyist.net/magazine/>

今後の情報源

Ruby リファレンスマニュアル刷新計画

- <http://doc.loveruby.net/refm/api/>

okkez weblog

- <http://typo.okkez.net/>