

GitHub を利用した Ruby 学習支援ソフトの開発

情報科学科 西谷研究室 2549 浦田 航貴

1 はじめに

Ruby は本格的なオブジェクト指向プログラムが記述できる汎用性の高い日本発のオープンソースである。Ruby は初心者に分かり易く、プログラム教育にもスムーズに活用できるメリットがある [1]。

西谷研究室に在籍している学生は、Ruby プログラミングを修得するために GitHub を利用した初心者向けの web 上の問題集を使って学習している。さらに、進捗状況の管理や指導者からの添削をより容易におこなえるように改善するため、バージョン管理ソフト GitHub を利用するシステム (rubynovice) を開発している。そこでは、Ruby プログラミングで重要となるテスト駆動をおこなえる環境を提供している。これにより、学習者自身が出力チェックできるようにし、Ruby プログラミングにおけるテスト実行に自然と慣れるような学習形態を目指している。しかし、テスト実行において、初心者が混乱する状況となっている。

そこで本研究は Ruby のテストフレームワークを変更することでこの混乱を回避することを目的にしている。

2 GitHub について

GitHub は、コンピュータープログラムの元となるソースコードをインターネット上で管理するためのサービスである。複数人が携わるソフトウェア開発において、ソースコードの共有や、バージョン管理といった作業は必要不可欠となる。またソースコードを始めとするプログラム開発に必要なファイルやそれらの変更履歴等を保存する「リポジトリ」と呼ばれる場所があり、ソースコード等のバージョンを管理する機能の他、プログラム開発等に対する開発者間でのレビューやコメント機能、プログラム開発の進捗を管理する機能等が備わっている [2]。

3 現状

3.1 テスティングフレームワーク

Rubynovice のテストフレームワークとしてはRSpecを実装している。これは、プログラムの振舞いを記述するためのドメイン特化言語を提供するフレームワークであり、「プログラムの振舞い」は、プログラム全体あるいは様々なレベルでの部分 (モジュールやクラス、メソッド) に対して期待する振舞いである。またドメイン特化言語 (Domain Specific Language:DSL) は、特定の問題領域 (ドメイン) を記述するために設計された言語である [3]。

RSpec によって期待されている値と出力している値が一致しているかを確認できる。テストコードを使えば「puts を使って毎回目視で確認」とするよりも、高速で確実に実行結果を検証することができる。

しかし簡単な Ruby のコードを書く場合、わざわざ RSpec 書くのは大げさであり、簡単なテストコード `assert_equal` が使えれば十分だと考えた。下記の表 1 は Test::Unit と RSpec を比較した表である。

表 1 Test::Unit vs RSpec.

	Test::Unit	RSpec
シンタックス	ピュア Ruby	DSL
修得ハードル	Ruby の文法	DSL を覚える必要あり
利用率, 情報量	低い, 少ない	高い多い
Ruby との関係	Ruby 標準バンドル	Ruby 標準ではない
本体のコード	少ない, シンプル	多い, 複雑

修得ハードルにおいて、Test::Unit は Ruby の文法なのに対して RSpec は、DSL (ドメイン固有言語) を覚えなければならないので、テストコードを作成するとき学習コストが大きい。また Test::Unit は、Ruby 標準バンドルであるという利点もある [4]。

4 結論と今後の課題

現段階では、西谷研究室に在籍している学生はテストする時、RSpec を使用している。また初心者として私自身が RSpec を使用したところ、少しの間違いでもエラーが複雑に表示されることや、すぐにエラーを理解することが困難であった。より単純でわかりやすい結果を出力してくれる Test::Unit (minitest) を用いることを考えている。

参考文献

- [1] 「Ruby 入門教育」, 池本有里, 山本耕史,
<http://www.shikoku-u.ac.jp/education/docs/Ser.A%20No.37,Ser.B%20No.34-20.pdf>.
- [2] 「GitHub」, 横田一輝,
<https://kotobank.jp/word/GitHub-1725201>.
- [3] 「Rubyst Magazine」, かくたに もろはし vol.54
<http://magazine.rubyist.net/?0021-Rspec>.
- [4] 「give IT a try」, 伊藤淳一, <http://blog.jnito.com/entry/2015/07/13/073458>.