

差出人: Naruaki TOMA <tnal@ie.u-ryukyu.ac.jp>
件名: [slab34-ec:2] JUnitの補足
日時: 2012年11月29日 20:05:12 JST
宛先: Evolutionary Computing <slab34-ec@ie.u-ryukyu.ac.jp>
返信先: slab34-ec@ie.u-ryukyu.ac.jp

出典: 「Java言語で学ぶリファクタリング入門」 <http://www.hyuki.com/ref/>
サンプルソースの場所:
hg clone ssh://info3@shark//home/info3/HG/2012/share2
cd JUnit

=====
(1) JUnit のインストール

1. <http://www.junit.org> からパッケージダウンロード。

11/27時点での最新版(安定版)は junit4.11.zip

2. 解凍し、適当な箇所に設置。以下は ~/javalib/ に設置する例。

```
prompt> cd ~/Downloads  
prompt> unzip junit4.11.zip  
prompt> mkdir ~/javalib  
prompt> mv junit4.11 ~/javalib/
```

3. CLASSPATHの設定

手順2 で設置した場所 (パス) を、
~/.zshrc にて環境変数 CLASSPATH に設定。上記例だと、以下のように追加。

```
----- ~/.zshrc への記入例 -----  
#CLASSPATH をexport してる箇所の最後に追加  
export JUNITDIR=~/.javalib/junit4.11  
CLASSPATH=$CLASSPATH:$JUNITDIR/junit-4.11.jar:$JUNITDIR/
```

4. CLASSPATH の確認

```
prompt> source ~/.zshrc  
prompt> echo $CLASSPATH  
*出力に ~/.javalib/junit-4.11/junit-4.11.jar が含まれていればOK
```

5. JUnit 自体の動作確認

```
prompt> java org.junit.runner.JUnitCore org.junit.tests.AllTests  
*出力の最後に OK と出ればOK
```

=====
(2) JUnit を使ったテストの流れ

1. テスト用クラスを作成。

- (a) テスト前に準備しておくべき手続きは @Before アノテーション付きで、
処理を済ませておく。サンプルでは setUp() で用意。
- (b) テスト後に必要な後処理を @After アノテーション付きで、
処理を済ませるようにする。サンプルでは tearDown() だが、中身は空。
- (c) 具体的なテスト項目を @Test アノテーション付きで列挙。

2. テストクラスをコンパイルし、JUnitを使ってテストを実行。

```
prompt> javac PersonTest.java  
prompt> java org.junit.runner.JUnitCore PersonTest  
*今回の例では全てのテストが通るはず。
```

=====
Q1: テストに失敗したらどうい出力になるだろうか？

Q2: 簡単な例だからというのがあるが、基本的に「関数/メソッドの実行結果を
検証する」というスタイルになっているのは何故だろうか？

Q3: PersonTest.java には main() が無い。何故実行できるのだろうか？

Naruaki Toma
E-mail: tnal@ie.u-ryukyu.ac.jp, Tel: 098-895-8830
<http://www.eva.ie.u-ryukyu.ac.jp/~tnal/>

--

You received this message because you are subscribed to the Google Groups "Slab34-ec" group.

To post to this group, send email to slab34-ec@ie.u-ryukyu.ac.jp.

To unsubscribe from this group, send email to slab34-ec+unsubscribe@ie.u-ryukyu.ac.jp.