

run-test.el

須藤功平

株式会社クリアコード
2009/09/056

今日のタスク



私とEmacs

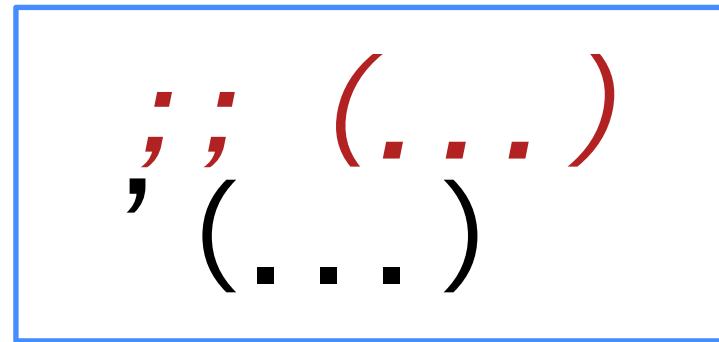
- ✓ 2000年～
- ✓ Mule + Mew + trr
- ✓ Viはたしなむ程度
- ✓ 書きものはほとんどEmacs

私とキー バインド

- ✓ 好きなバインド: `¥M-:`
- ✓ プレフィックス: `¥C-c`
- ✓ `¥C-cX`より `¥C-c¥C-X`

私とELisp

- ✓ run-test.el: 開発支援
- ✓ rabbit-mode: Rabbit支援
- ✓ コメントは'



;; (...)
'(...)

あなたとEmacs?



自由な
エディタで
自由な
ソフトウェアを

ソフトウェア

- ✓ 作りきり
 - ✓ 一発ネタ
 - ✓ プロトタイプ
- ✓ 作って、よくして、保守
- ✓ 使いづけるソフトウェア

続けること

- ✓ 価値のこと
 - ✓ Emacs: 1976～: 33年！
- ✓ 大変なこと
 - ✓ XEmacs
 - ✓ 21.1 → 22.1: 約6年！

障害

- ✓ リソース不足
- ✓ やる気不足
- ✓ さわりたくないコード

火

やつつけ

——ト”

!=
!

ハツク

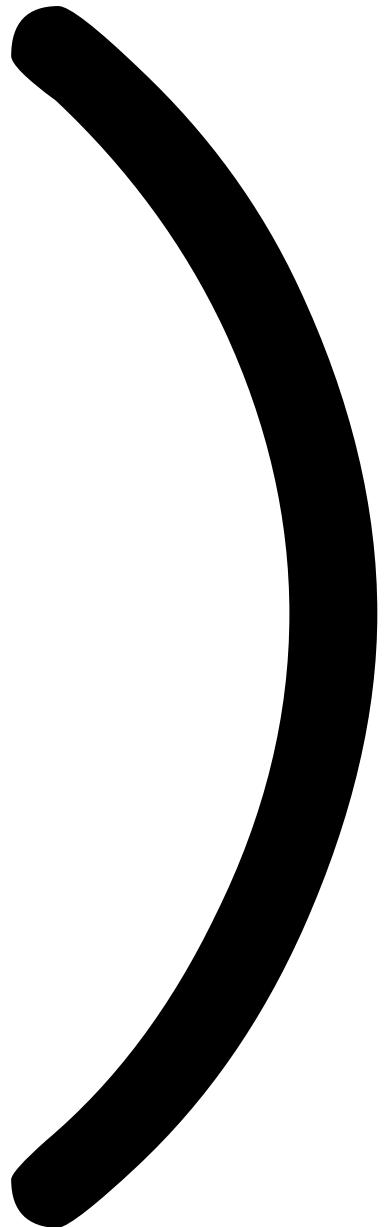
ハック

開発が続けられるコード

12/28

ハッカー

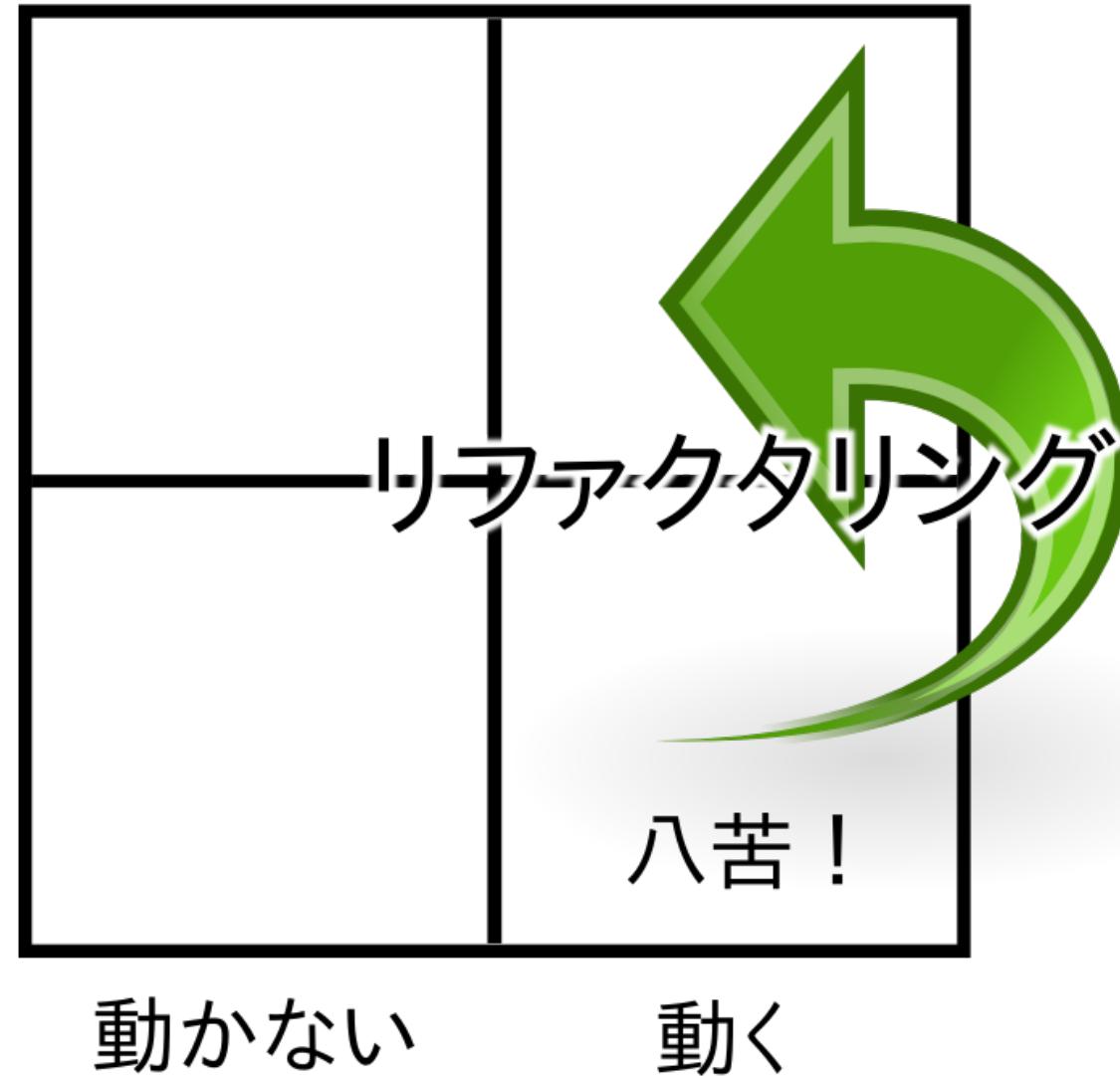
通った後に道ができる
それがハッカー



きれいへ

きれい

汚い



リファクタリング

- ✓ 動作は変えずにきれいに
- ✓ 変更→テスト→変更→テスト→
変更→テスト→変更→テスト→
...
- ✓ 動く！きれい！

run-test.el

それっぽいテストを起動

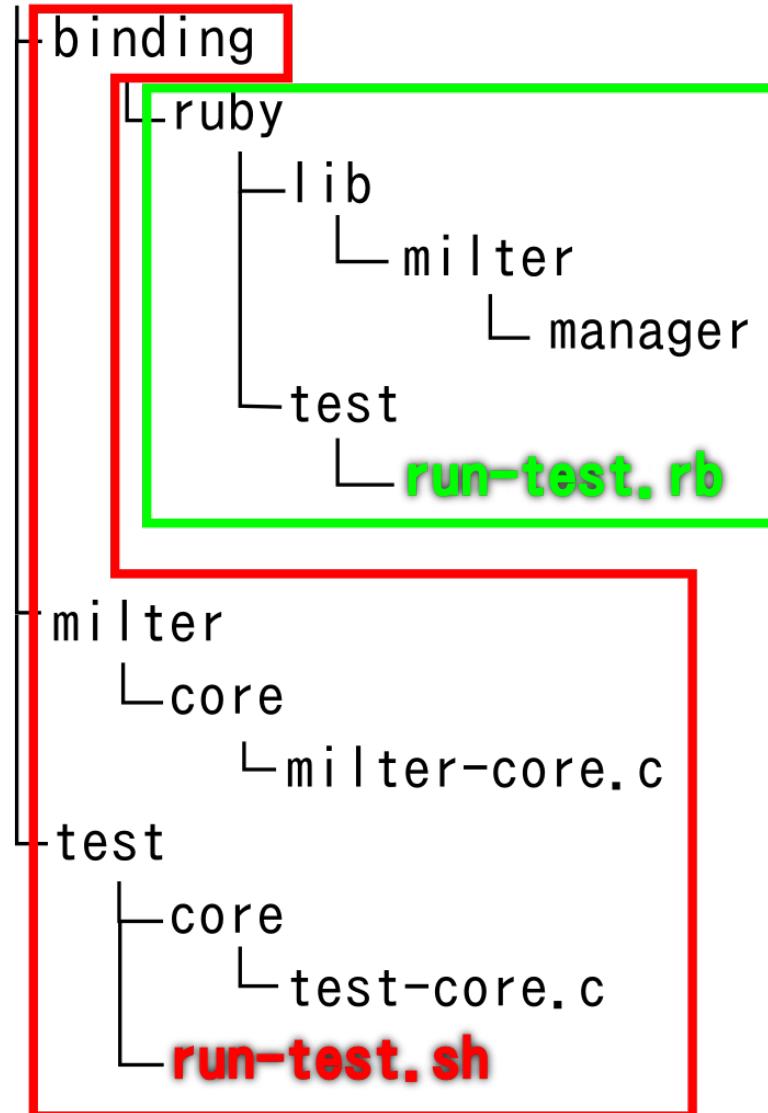
ツリー

```
binding
└ ruby
    └ lib
        └ milter
            └ manager
    └ test
        └ run-test.rb

milter
└ core
    └ milter-core.c

└ test
    └ core
        └ test-core.c
    └ run-test.sh
```

それっぽい



デモ

スタックを作ろう

20/28

test-make

```
(define (test-make)
  (define stack (make-stack))
  (assert-true (stack-empty? stack))
  #f)
```

make & empty?

```
(define (make-stack)
  ' (stack))
```

```
(define (stack-empty? stack)
  (null? (cdr stack)))
```

test-access

```
(define (test-access)
  (define stack (make-stack))
  (stack-push! stack 1)
  (assert-false (stack-empty? stack))
  (assert-equal 1 (stack-pop! stack))
  (assert-true (stack-empty? stack))
#f)
```

push! & pop!

```
(define (stack-push! stack value)
  (set! (cdr stack)
        (cons value (cdr stack))))
```

```
(define (stack-pop! stack)
  (let ((value (cadr stack)))
    (set! (cdr stack) (cddr stack))
    value))
```

push! & pop!: setter

```
(define (stack-push! stack value)
  (push! (cdr stack) value))
```

```
(define (stack-pop! stack)
  (pop! (cdr stack)))
```

まとめ

マグカップは忘れずに！

おすすめ

run-test.elで
開発を続けられる
ソフトウェアを書こう

10月はたいやきツアー



28/28