

とある道場の
乱取稽古
TDD Dojo

TDD 道場

～ みんな TDD やってみよう! ～



わんくま同盟 名古屋勉強会 - TDD 道場

TDD 道場 ～ 乱取り

- プロジェクトに接続された 1台の PC でコーディングする。
- ペアでコーディングする。
- **5 ～ 10分間隔でペアの片方を交代していく。**
聴衆 ⇒ ナビ、ナビ ⇒ ドライバー、ドライバー ⇒ 抜ける
- **コーディングを担当**しているときは、自分が何をしているのかを**説明しながらキーボードをタイプ**する。こうすることで聴衆も、何が起きているのかを理解できる。
- ~~• **聴衆**は、テストが **GREEN** の場合にだけ、設計について意見を述べてもよい。テストが **RED** の状態では、設計については質問しかできない。(RED の時に助け舟を出していいのは、アドバイザーのみ)~~
- ~~• **聴衆**が今行われている作業について混乱してきたら、コーディングしている人は手を止め、今やっていることを説明する。~~



ペアプログラミング

- ドライバー
 - キーボードを打つ
 - 目の前に集中
- ナビゲーター
 - 先を見る / 周囲に気を配る
- ペア プロの極意
 - 会話し続けること !!

ローカルルール

- 交代は 5分を目安
 - テストケースを 1つ書き、RED⇒GREEN まで
 - 時間が余ったら、リファクタリング
- 最初に口頭でテストケースを宣言する
 - 「〇〇という入力するとき、△△になるケースを書きます」と宣言してから、テストケースを書き始める。

今回のルール

- 交代時間を気にせずやってもらいます
 - また、会場からも、どんどんアドバイスを出してください。
 - まだ3回目だというのに、ぜんぜん「道場」してなくてすみません m(_`_)m

名古屋#13 の お題

- 時刻に応じた挨拶 by C# & NUnit
 - 午前中(5:00~12:00)は "Good morning!"、それ以外は "Hello!" と返すメソッドを作る。
 - 製品コード: Greeting クラス
Say() メソッド
 - テストコード: GreetingTest クラス
ひとつめのテストケース:
SayTest1_午前中_GoodMorning()
※ ここまで準備済み

道場 スタート

- テストケースは、みっつ
- 一つ目のテストケースは、簡単に通せる!
"Good Morning" と返すだけ
- 二つ目以降は、…どうしましょう!?
※ NMock は入ってます



外部 I/F を分離する

- 外部との I/F 部分は分離すべし
 - システムクロック、DB、ファイル、etc.
 - テストしやすいだけでなく、変更強い設計にもなる (はず)。
- 分離するには?
 - インターフェースを使って、差し替え可能に。
 - #if ディレクティブで強引に切り替え。
 - ダミー製作が面倒なときは Mock を。



附録: TDD 三原則

<http://yattom.jp/trac/public/wiki/TDDByUncleBobMartin>

- RED ⇒ GREEN

失敗するユニットテストを成功させるためにしか、プロダクトコードを書いてはならない。

- 失敗する(と思われる)テストケースだけ

失敗させるためにしか、ユニットテストを書いてはならない。コンパイルエラーは失敗に数える。

- テストをギリギリ通るだけ

ユニットテストを1つだけ成功させる以上に、プロダクトコードを書いてはならない。





わんくま同盟 名古屋勉強会 - TDD 道場