最速でJavaエンジニアになって稼ぐ

2018/12/06

人生逃げ切りオンラインサロン内

目次的なやつ

- アプリケーションのコード解説
- 自動テストとは
- JUnit実践

アプリケーションのコード解説

GitHubのコードを見ながら
一つ一つ解説。

日付計算アプリケーションのサンプル
https://github.com/koujienami/DateCalculation

Spring Boot解説(基本編：Controllerとは)
https://qiita.com/TEBASAKI/items/267c261db17f178e33eb

ドメイン駆動設計
https://ja.wikipedia.org/wiki/%E3%83%89%E3%83%A1%E3%82%A4%E3%83%B3%E9%A7%86%E5%8B%95%E8%A8%AD%E8%A8%88

自動テストとは

名の通り「自動」で「テスト」をするもの。
でも何もしなくてやる、という意味ではない。

- テストというのはアプリケーションや特定の機能、クラス、メソッドが想定している通りの振る舞いをしているかを確認する為のモノ。
- その為、アプリケーションが動作してから、実際に動かして確認する事も多い。クラスなどの場合は、実際にそのクラスやメソッドを別のクラスから呼び出したり、mainメソッドを作ってそこから動かしてみたりして、実行確認をする。
- しかしmainメソッド経由で呼び出すクラスはそのまま残しておけない。テストが終わったら消すといった作業が必要になる。別のプロジェクトで管理するのも手。でも管理が煩雑。どうしよう。
- そこで利用されるようになったのが「テストフレームワーク」と呼ばれるもの。

- Javaにおいて「テストのフレームワークは何を使ってますか？」と聞かれた時に一番槍に上がるほとんどが「JUnit」。1997年に開発され、現在では最早デファクトスタンダードとなり、使われない事の方が珍しい。
- JUnitでは、特定の処理の呼出や事前準備などを楽にする機能が大量に備わっており、またJUnit以外のフレームワークやライブラリでも機能拡張されているのでテストに必要なことはほぼ全て揃っている状態。
- 対象としては「単体テスト」が主なのでクラス単体、機能単体のテストが得意。でも結合レベルや総合レベルが出来ないわけではない。
- 素早くテストが作成でき実行も早く、誰でも同じテストが実行でき、独自技術のテストコードではないので流用ができ、一度作れば何度でも実行でき、テストコードを見れば仕様が一目瞭然という状態にも出来る。
- テスト駆動開発を行う場合はJUnitでテストコードを書き、それを仕様としてプロダクトコードを作成する。

JUnit実践

日付計算アプリケーションにテスト
コードを追加する。

日付計算アプリケーションのサンプル
https://github.com/koujienami/DateCalculation

Spring Bootでテストを書くときのやりかたまとめ
https://qiita.com/mitsuya/items/be50dc329b4f3abe5ac5

Spring Bootとユニットテスト環境の設計について
https://qiita.com/rubytomato@github/items/0baf1df5b3eb7094cc41

mybatis-spring-boot-starter 1.3の変更点
https://qiita.com/kazuki43zoo/items/3f94d2049d331f81e6f5

次回までの宿題

作成したアプリケーションに
テストコードの追加。

おしまい