

成果発表メモ_0704

0. 自己紹介

- 名前：永倉佑一（ながくらゆういち）
- 年齢：参加申し込み条件ギリギリ
- 血液型：A型
- 勤め先：NTTデータ(ヒラ社員)
- 最近やっている仕事：
 - 最近は機械学習で自然言語処理をメインでやってます。
 - Slerの仕事も減ってきたりしているので、仕事取るために提案に行ったり
 - 既存の情報系システムのマイグレーション手伝いながら、やりかたを少しずつ今風にしたりしてます（15年ぐらい前のシステムなのでとっても古い）
 - インフラとかプロジェクト管理（進捗、調整）とかの方がメイン。
- 触れる言語：
 - Ruby
 - Python

1. 応募動機

SparkをScalaで使えるようになりたかった

- バックエンド処理のフレームワークとして、Spark(OSS)を使ってアプリケーションをScalaで書けるようにそろそろちゃんになりたいと思った。

- お恥ずかしい話ですが...実装はオフショア先に任せてしまっていてScalaチックなコードになると、Rvがとっても怪しくなっている現状。
- 弊社内では、スケールアウト前提のアプリケーションはSparkで考えるお客さんも多くなっている感じ。

Scalaへの憧れ

- 自分の周りのScala書ける人が優秀な人多い（気がする）。
- 年収も高くて、これから需要が高くなって、意識高い系の人材になれると聞いた。

2. できたところまでの課題

できたところまで

Lesson14まで。（の課題は全部提出できた。） Lesson15のMicroPostの課題で断念。

工夫した点

工夫をする余裕が残念ながらなかったです。

3. 気づいた点や学んだ点

Scalaについて

感動したり、取り組みを通じて気に入ったりした順です。

いい印象について

- Scalaがコンパイルせずにスクリプトで起動できることにビックリした。小さなものから大きなものまでのスケーラブルという言葉がしっくりきた。
- caseクラスがいろいろやってくれていることに震えた。

- コンパニオンオブジェクトを自動で作ってくれる
- コンストラクタのパラメータに、valを付けなくてもフィールドが自動生成される
- equals, canEqual, hashCode, toString, copyメソッドを自動生成
- コンパニオンオブジェクトにapplyメソッドとunapplyメソッドを自動生成
- applyメソッドでnew演算子を使わずにインスタンス生成が可能
- unapplyメソッドでクラスの構造を知らずとも属性に分解できる

- キレイに完結に書けるようになった(気がする)
 - いつもよく書いてしまうようなコード (Python)

```
# pythonで配列の文字の長さの合計を求める
strLen = 0
strList = ["hoge", "fuga", "bar"]
for (i in strList):
    length = len(i)
    strLen += length
```

- 受講後のScalaコード

```
# scalaで配列の文字の長さの合計を求める
val strList:Seq = Seq("hoge", "fuga", "bar")
strLen = strList.map(_.length).foldLeft(0)(_ + _)
```

Scalaで理解に苦しんだところについて

1. ジェネリクス。Javaに経験がないこともあり、ジェネリクスの共変、非変、反変、上限境界、下限境界はとても難しく感じた。
2. PlayFramework全般。同じものを作るにしても、Railsのほうが書きやすいと感じてしまった。

カリキュラムについて（感想）

- 1ヶ月のボリュームにしては、多く感じましたが、色々網羅されていて、課題マイルストーンもあって、他にはない後生、参考にできそうな教材と思いました。
- Lesson5の映画館の課題で何度も再提出になりましたが、Rvを繰り返し行なっていく過程でリファクタリングしていく作業は実務に即して面白かったです。実務の観点でコメントいただけたことが非常に有益でした。（不出来な課題提出ばかりで申し訳ありませんでした。Rvいただきありがとうございます。）

その他

個人的には、PlayFrameworkよりもLesson10までの課題をもっと増やして、WEBフレームワークではない、実務よりのスキルを身に付けられるようなカリキュラム（例えば、バッチ処理をScalaでどうやって書くといいか、ScalaTestについての解説がもう少しあってほしいなど）があったら、尚嬉しいです。