

今週の一コマ画像

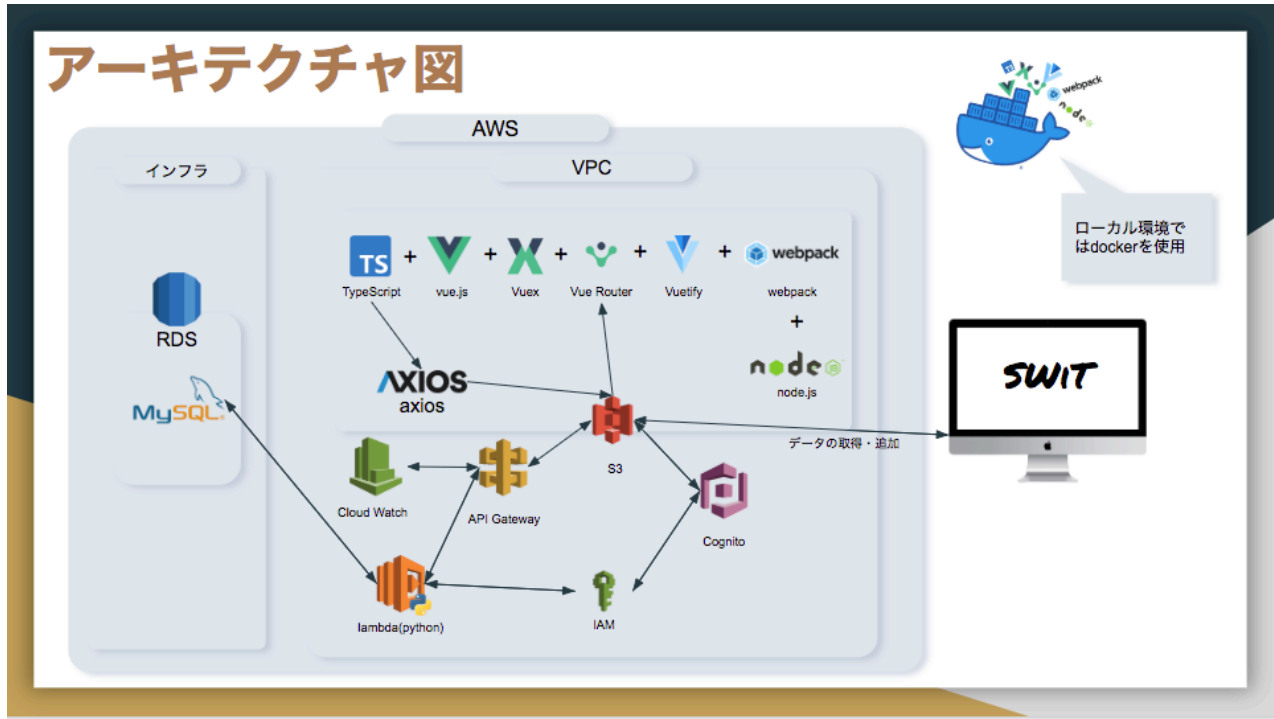


図0...先日ハッカソンで作成したアーキテクチャ図。満足の出来。

応用プログラミングJava

宿題レポート APJ01

学籍番号: 744366

氏名: 岸 典樹

2020年9月27日

第一章 「HELLO WORLD.」 についての考察	3
1.1 発展プログラミング演習で行った 「Hello world.」	3
1.2 青木先生作の 「Hello world.」	5
1.3 2つの 「Hello world.」 の比較	9
1.4 演習科目 「Hello world.」 と青木先生の 「Hello world.」 についての考察	9
第二章 「APACHE ANT」 を使用する意図について	10
2.1 Eclipseとは	10
2.2 Apache Antとは	10
2.3 Apache Antを使う教育的な意味	10
2.4 考察	10
第三章 自由記述	11
3.1 まずはじめに...	11
3.2 Javaに関する雑談	11
3.3 参考文献	13

第一章 「HELLO WORLD.」 についての考察

本章では発展プログラミング演習で行った「Hello world.」と青木先生作の「Hello world.」について共通点と相違点を分析し、動くプログラムではなく正しいプログラムを作るという観点から、自らが採用しなければならないことが何であるのか考察する。

1.1 発展プログラミング演習で行った「Hello world.」

以下に、発展プログラミング演習(以下基礎科目)で行った「Hello world.」を記載する。

```
1. public class HelloWorld { // <= クラス宣言の開始
2.     public static void main(String[] args){ // <= mainメソッドの開始
3.         System.out.println("Hello Wrold");
4.     } // <= mainメソッドの終了
5. } // <= クラス宣言の終了
```

図1.1.1 発展プログラミングで行った「Hello world.」(HelloWorld.java)

基礎科目では上記のHelloWorld.javaを以下のコマンドでコンパイル・実行を行う。

```
$ javac HelloWorld.java
```

```
$ java HelloWorld
```

[実行結果]

```
> Hello World
```

図1.1.2 基礎科目のHelloWorld.javaの実行結果

上記プログラムはJavaにおける「Hello world.」の最小構成である。さらに、それをjavacコマンドでコンパイル、javaコマンドで実行することでコンソールに文字列を出力するものであり、初学者向けの動くプログラムと言える。

1.2 青木先生作の「Hello world.」

青木先生作の「Hello world.」は以下のディレクトリ構造で構成されている。

```
$ tree
$ .
$ └─ Classes
$ |   └─ example
$ |       └─ Example.class
$ └─ Example.app
$ |   └─ Contents
$ |       └─ Info.plist
$ |       └─ MacOS
$ |           └─ applet
$ |       └─ PkgInfo
$ |       └─ Resources
$ |           └─ Java
$ |               └─ example.jar
$ |       └─ JavaApp.icns
$ |       └─ Scripts
$ |           └─ main.scpt
```

図1.2.1 青木先生作の「Hello world.」のディレクトリ構造1

```
$ |           └─ applet.icns
$ |           └─ applet.rsrc
$ |           └─ description.rtf
$ |           └─ TXT.rtf
$ └─ HelloWorld.png
$ └─ Makefile
$ └─ build.xml
$ └─ example
$ |   └─ Example.java
$ └─ example.jar
$ └─ example.mf
$
```

図1.2.2 青木先生作の「Hello world.」のディレクトリ構造2

以下に、青木先生作の「Hello world.」を記載する。

```
1. package example;
2.
3. /**
4.  * 例題プログラム：この例題を改変して大きなプログラムを作る足がかりにしてください。
5.  */
6. public class Example extends Object
7. {
8.     /**
9.      * 例題のメインプログラム。
10.     * @param arguments 引数の文字列の配列
11.     */
12.     public static void main(String[] arguments)
13.     {
14.         System.out.println("Hello world.");
15.
16.         return;
17.     }
18. }
```

図1.2.3 青木先生作の「Hello world.」 (example/Example.java)

青木先生作の「Hello world.」ではbuild.xmlによって設定されたAntによってビルドされる。今回の場合、ant関連コマンドはMakefileによって管理されている。以下に、実行方法を記す。

```
$ make install # アプリケーションのビルド  
$ open Example.app # アプリケーションの実行
```

図1.2.4 青木先生作の「Hello world.」のビルドと実行

make installによって生成されたExample.appを実行した結果が以下である。

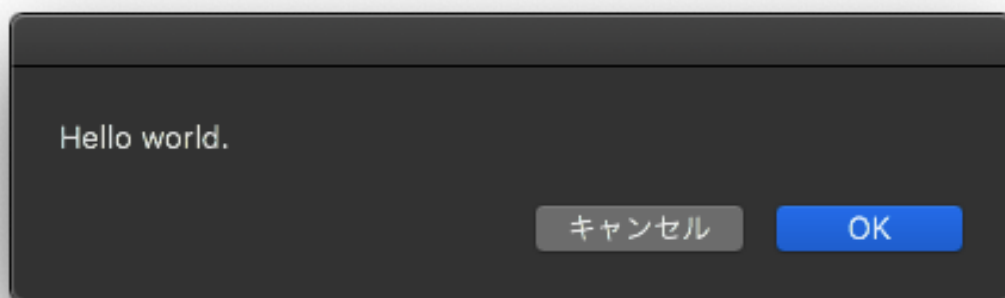


図1.2.5 青木先生作の「Hello world.」の実行結果

こちらのプログラムは演習科目の「Hello world.」と違い、アプリケーションファイルにビルドすることでポップアップするようだ。次項で、これら2つのプログラムを比較し、共通点と相違点を分析していく。

1.3 2つの「Hello world.」の比較

まず、javaプログラム本体から比較を始める。

演習科目のHelloWorld.javaに対して青木先生作のExample.javaは概念的には同じプログラムであっても以下の点で違いが見られる。

- ・ パッケージが宣言されている
- ・ クラス階層のルートであるObjectを継承している
- ・ javadoc用のコメントが記述されている
- ・ void型でもreturnを宣言している
- ・ 左プレスの前に改行している(これは作法があるかもしれない)

次に、javaプログラム以外の部分を比較していく。青木先生作の「Hello world.」はクラスをエクスポートするディレクトリ、アプリケーションファイル、ビルド関連ファイルなどが設定されており、ディレクトリ構造が整理されているものとわかる。さらに、java本体を含めたUMLを添えることでプログラム全体の構造がGUIとして認識できる形となっている。

次項で2つのプログラムを比較した結果をまとめ、考察する。

1.4 演習科目「Hello world.」と青木先生の「Hello world.」についての考察

演習科目の「Hello world.」は初学者にその言語の「概念」を教えるための「動くプログラム」であり、それ以上の目的はないため最小限の規模で構成されている。一方、青木先生作の「Hello world.」はスケラビリティに溢れており、大規模プロジェクトの礎となっている。昨今、プログラマにはスピード感のある開発が求められるようになってきた。個の技術を向上させることはもちろんであるが、拡張性や可読性を考慮したコーディングやビルドなどの自動化は誰もが利用できるものであり、チーム開発において重要な要素のひとつである。一般に、プログラムはひとりで作る場合は極めて限られているため、社会でプログラムを書く人はこれらにより優れたプログラムを書くことが求められる。これはソフトウェア工学2の「ひとりじゃないって」に強く関連するのではないかと考える。

第二章 「APACHE ANT」を使用する意図について

本章ではJavaプログラミング環境としてEclipseではなくApache Antを利用する教育的意味について考えていく。

2.1 Eclipseとは

Eclipseとは、Java, C++, C#などの言語のIDEであり、主にアプリケーションの開発に用いられる。コーディング以外のほとんどの作業はGUI上で行えるため、効率良く開発する場合有力な選択肢のひとつである。しかし、複数人で開発を行う場合、全員がEclipseを導入しないとEclipseの恩恵を受けられないことが弱点である。

2.2 Apache Antとは

Apache AntはJavaプログラムをビルドするツールであり、それ単体で動作することができる。Eclipseの場合先述の通り全員がEclipseを導入し、その上でビルドできないといけませんが、Apache Antはバージョン管理システム等で管理されるbuild.xmlの設定情報とApache Antの環境さえあれば誰もが同じ恩恵を受けることができる。

2.3 Apache Antを使う教育的な意味

なんでもGUIをポチポチするだけで動作するEclipseに比べ、ビルドに関する設定を自ら調べて書くこと、その中でビルド後のバイトコードをどのディレクトリに展開するかなどのシステムを作る上でより根幹となる箇所の勉強になることが考えられる。また、それらをmakeコマンドで自動化するように設定すれば、作業の自動化を行解することができる。

2.4 考察

個人的に、Apache AntはDocker(docker-compose)に似た性質を持つものだと考えている。build.xml(docker-compose, Dockerfile)にそれぞれの設定を書き、それらをコマンドで呼び出すことで環境に依存せず、意図した動作を可能にする。近年はkubernetesなど自分で書いたコードを共有することですべての人が同じ恩恵を受けられるようになってきているので、2.3節が正しいのであれば、今後Javaで開発を行う際にはぜひ選択肢に入れ、学びたいと思う。

第三章 自由記述

3.1 まずはじめに…

1年半の短い期間(と信じていたのですが)改めて、よろしくお願いします。今学期は応用プログラミングJava、応用プログラミングPython、プログラミング言語、特別研究とレポートでお会いする数が非常に多くなると思いますが、頑張っていきたいと思います。今週は、少し余裕があったのでソフトウェア工学2に比べ、レポートに割り当てる時間を2倍にしてみました。時間が許す限り維持していきたいと思います。オンライン飲み会、楽しみにしてます！

3.2 Javaに関する雑談

学生が気が気でなかった春学期の成績ですが、(もしかして)ちょっと早く見たら何か面白いことになってるんじゃないかと思って閲覧してみました。(次ページへ続く)



図3.2.1 webskまさかの表示

いろいろ突っ込みどころはありますが、まず、フロントは新しくしたのにバックはservletのままなんですわ...

よく見ると、オブザーバモデルを使っている点も確認できます。うーん...ここまで表示してよかったんでしょうか(笑)

成績開示前には元に戻っていたのでよかったのですが、ちょっと面白くて友人らと笑ってました。

3.3 参考文献

- ・ クラスObject

<https://docs.oracle.com/javase/jp/8/docs/api/java/lang/Object.html>

- ・ 目次の作り方

<http://epub.jugem.jp/?eid=41>

- ・ 演習科目の「Hello world.」

<https://ksuap.github.io/2018spring/lesson01/basic/#hello-world>

- ・ 青木先生作の「Hello world.」

<http://aokilab.kyoto-su.ac.jp/repositories/AP/Java/Lectures/index-j.html#HelloWorld>

- ・ Pagesでプログラミングのコードを綺麗に書きたい

<https://amitica.net/archives/596>