

フレームワークとは

- ・一般的な開発で利用される機能群や骨組みを提供してくれるもの
- ・開発を簡単にするためのソフトウェアの土台
- ・特定のプログラミング用語で書かれた半完成品のソフトウェア

フレームワークの中に含まれているもの

- ・さまざまな開発で使える、再利用可能な機能(クラス、ライブラリ、モジュール、APIなど)
- ・そのまま利用できる主要部分のひな形(テンプレート)
- ・開発者が機能を追加するための仕様や規約



これら既存の機能に加えて、開発者がさらに機能を追加することにより、目的のソフトウェアをより早く完成させることができる

フレームワークを使うメリット／デメリット

メリット	<ul style="list-style-type: none">・よく使う機能は既に用意されている・開発期間を短縮化・品質向上・ミスが減る・チーム内で共有できる・考え方の統一ができる
デメリット	<ul style="list-style-type: none">・フレームワークの使い方を学習する必要がある・不具合が起きたとき対処しづらい・開発を1から行う場合と比べ、技術力の向上は期待できない

【Spring Framework】(読み: すぷりんぐふれーむわーく)

- ・「Spring」とも呼ばれる
- ・JavaのWEBフレームワークで、WEBアプリを作ることができる
- ・万能なフレームワークなのでさまざまなシステム構築に対応可能
- ・従来のWebアプリケーションにおけるさまざまな問題を解決している



～Spring Frameworkの特徴～

1. 変更に強い

- ・Javaを使ったアプリ開発は大規模なものが多く、途中で修正や変更が入ることも多い
以前までは**変更に弱く、変更にかかなりの時間を費やす必要があった**

しかし

Spring Frameworkには「**DI(Dependency Injection) = 依存性の注入**」と呼ばれる仕組みが導入されている
これにより、依存性のあるプログラムは外部から取り入れるようになっているので、
ひとつひとつのプログラムを独立させやすくなる



最低限の変更で済み、開発が楽である

2. テストが簡単

- ・「**Spring MVC Test**」という専用のテストプログラムを使用できる
ひとつひとつの処理がうまくいっているかのテスト(単体テスト)ではなく、
無事に一連の処理を終えるかどうかのテスト(結合テスト)をすることができる



バグの混入を最低限におさえ、安全に開発をすすめられる

3. 拡張性が高い

- ・必要な機能だけを外部から使うことができる(無駄がない)
- ・必要な機能があるときは、あとから簡単に追加できる
- ・**オブジェクト指向の基本原則を徹底**して、Javaを最大限に生かすことができるよう設計されている

4. 保守性が高い

- ・「**AOP(Aspect Oriented Programming)**」というプログラミングを用いることができる
(クラスには「**本質的な処理**」のみを書いて、「**本質的ではない処理(共通化できる処理)**」を別を書く)



コードを分かりやすくする効果や
バグや不具合の原因を特定しやすくなる効果がある



5. 再利用性が高い

・「AOP」の仕組みにより、**共通のプログラムをまとめやすくなる**ので、再利用することも容易

プログラムを再利用する = プログラムを書く量が減る



修正する場合も**最低限の修正で済む**(長期のアプリ開発でものすごく効率に差がつく)

～Spring Frameworkで作る最適アプリ～

・WEBシステム



柔軟性や拡張性が高いので、
大規模な開発や仕様変更に対応しやすい

・WEBサービス



どんどん規模が大きくなっていくWEBサービスでも、
機能がまんべんなく提供されているので、柔軟に対応できる



【Play Framework】(読み: ぷれいふれーむわーく)

- ・「Play」とも呼ばれる
- ・プログラミング言語の「Java」と「Scala」を使って作られた
- ・Play Frameworkを使うと**アプリを高速で開発**することができる



～Play Frameworkの特徴～

1. 高速開発

- ・Play Frameworkは**軽量なフレームワークで動作も速い**
 - ・MVCの設計に基づいて開発を行うことができるので、WEBアプリを高速開発できる
 - ・弱点であった「コンパイルの遅さ」も解消されているため、**コードの追加や修正が即座に反映**される
- ※Javaはコードの実行前にコンパイルする作業が必要だが、毎回コンパイルすると時間をかけてしまう

2. リソース消費が少ない

- ・CPUやメモリーに負担が少なく、**少ないリソース(動作の実行に必要な処理システムの要素)でアプリを動作**できる

少ない資源で動く
無駄がない／速い



大規模まで耐えられるアプリを開発できる
将来のことをあまり気にせず開発できる

公式サイト: <https://www.playframework.com/>

3. 高い拡張性

- ・最新技術に関するものまで、幅広く対応している
- ・CSSの拡張言語であるSassや、JavaScriptの拡張言語であるCoffeeスクリプトにはじまり、**リアルタイム通信やテストに関するものまで一通り揃っている**



どんな開発でも**最も効率の良いものを選ぶことができる**
開発をスムーズに行うことができる

～Play Frameworkで作る最適アプリ～

- ・WEBシステム
- ・WEBサービス



費用の制限がある企業向けのシステムやスピードを求められるWEBサービスでは、**高速開発できるのが最大のメリット**



公式サイト: <https://www.playframework.com/>

【Java EE】(読み:じゃばいーいー)

- ・正式名称は「Java Enterprise Edition」
- ・**企業向けのWebアプリケーション開発用の機能をセットにしたフレームワーク**
- ・「Java SE」にWebアプリケーションの開発に必要な
サーバー関係のライブラリなどを追加したもので、
Java EEの中には**Java SEも含まれている**
(Java SEのライブラリを使うことができる)
- ・最新版は「**Java EE8**」(2017年8月現在)



～Java EEの特徴～

1. Java EEのSDK

- ・フルプラットフォームプロファイル(通常版)とWeb Profileの2種類がある
- ※Web Profile...フルプラットフォームプロファイルのSDK
(ソフトウェアを開発するために必要な技術文書やツールなど一式)の中から
一部のAPI(ソフトウェアの機能を共有すること)だけ抜き出した軽量化されたSDK
- ・すべての機能を使うには、フルプラットフォームプロファイルを選択する

2. Glassfishの同梱

- Java EEのSDKには、Glassfishというアプリケーションサーバーが同梱されている
- Java EEで開発したWebアプリケーションを動かすにはJava EEに対応したサーバーが必要



Glassfishを使えば、開発やテストを行うことができる

～Java EEの使用例～

- Webサイト(もしくは、Webアプリケーション)
- SNS
- ECサイト
- 予約サイト
- 銀行

