

課題 フレームワーク

1. Spring Framework

特徴: 汎用性及び拡張性が高い。環境に依存する機能を外部から提供可能とする。
DI(Dependency Injection:依存性注入)、AOP(Aspect Oriented Programming:アスペクト指向プログラミング)という機能。

DIはオブジェクトが他のオブジェクトを利用する「依存性」のあるコードを、実行時に結合する設計思想。依存性を排除することでコードの簡素化、テストの簡易化、コード改修・移行の簡易化が見込める。

AOPは、クラスには本質的な処理のみを書き、共通化出来る処理を別に書く事が出来る機能で、管理・変更・追加を容易にすることができる。またそれにより元のソースコードに変更を加えず新たな処理を追加出来る。

2. Play Framework

特徴: JavaとScalaを使って作られたWebフレームワークで、軽量さと高い生産性に特徴がある。CPUリソースやメモリ使用量が少ないので軽快に動作し、JavaEEに準拠しない代わりにシンプルに開発を進められる。

また、Javaの基礎的な知識があれば使い始められる事から学習コストも抑えられる。

その軽量さと高い生産性から、WebシステムやWebサービスを高速で開発出来ることが何よりのメリットである。

3. JSF(Java Server Faces)

特徴: JavaEEという環境で動作する標準的なコンポーネント指向(機能毎に部品として扱う考え方)のユーザーインターフェースフレームワーク。

ユーザーインターフェース(ユーザーが操作する見た目の部分)をかんたんに作ることが出来るフレームワークで、MVCモデルが採用されている。

開発チームとデザインチームで分業出来るメリットがあり、スタンダードに使われている技術なので利用する事も多い。

4. JavaEE(Java Platform, Enterprise Edition)

特徴: Java標準仕様のフレームワーク。Oracle社公式のフレームワークであり、参考文献が多かったり認定資格があるなど、学習がしやすいフレームワーク。

フルスタックフレームワークといい、ServletやJSP、Javaの各種APIなど、大規模システムの構築に必要な機能がまとめて提供されている。

ただし大きなフレームワークだけに更新頻度は高くなく、最新技術を反映させるよりも標準仕様としての安定感が重視されている。