

# Spring Framework

○どういったメリットがあるのか？

## ①変更に強い

Spring Frameworkでは、DI(Dependency Injection)と呼ばれる仕組みが導入されている。

(日本語で言うと依存性の注入という意味)

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が、**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、1つ1つのプログラムを独立させやすくなる。**

そのため、**変更点がある場合も最低限の変更で済むので開発がスムーズになる。**

## 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うことができる。

これは、1つ1つの処理がうまくいっているかをテスト(単体テストと呼ばれる)するのではなく、無事に一連の処理を終えるかどうかをテスト(結合テストと呼ばれる)することができる。

それによって、**バグの混入を抑え、安全に開発を進めることができるようになっている。**

## ③拡張性が高い

フレームワークの基本的な機能の大部分が「インターフェース」として提供されているが、必要な機能だけを外部から使うことができるので無駄がない。

また必要となる機能がある場合は後から追加することも容易にできる。

**オブジェクト指向の基本原則を徹底して生かすことができるように設計されている。**

## 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。

これは、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)」を別に描くという方法。

主に**本質的な処理とそうでない処理を分けることで、コードをわかりやすくする効果がある。**

バグや不具合があった場合の原因を特定しやすくなる。

## 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ため、再利用が容易になる。**

再利用により**プログラムコードの量が減り、修正する場合も少ない修正で済むようになる。**

再利用性の向上をプログラマーの腕ではなく仕組みで解決している。

上記記述参考先：

<http://www.sejuku.net/blog/10456> (5分でわかるフレームワークSpring Framework)

# SpringFrameworkを使用するために理解すべき事

## Dependency Injection : 依存性の注入とは？

上記では直訳で「依存性の注入」と説明したが詳しく見ていこう。

DIという言葉のうち「**Dependency (依存性)**」という単語は、「**オブジェクトが成立するために必要な要件**」という意味を持っている。

この要件とは、オブジェクトが持つ属性や関連するオブジェクト等の事である。

例えばパソコンクラスにOS=Windowというフィールドがあり、モニタがパソコンクラスに利用関係を持っている。この場合モニタークラスやOSフィールドがパソコンクラスの依存性となる。

次の単語である「**注入(Injection)**」とは「**外部からの設定(Configuration)**」を意味する。**設定ファイルやWebアプリケーションのデプロイメントディスクリプタ(web.xmlなど)での設定を「注入」と呼んでいる。**

例えば:

ClassAにValueという抽象化された変数がある。ここに設定ファイルを外部から設定する。これが注入である。

これらの事から「DI」という言葉を言い表すと「オブジェクトの成立要件に必要な情報を外部設定すること」となる。**情報を外部に切り出すことでオブジェクトを利用する状況が変わったとしても、設定を変更するだけでそのオブジェクトを利用することができるようになるので、再利用性がとても高い「部品」としてのオブジェクトを実装しやすくなる。**このような再利用性の高いソフトウェア部品の事を「コンポーネント」と呼ぶ。

上記の説明のようにDIの考え方はJavaというオブジェクト指向ととても似ているため、開発における利用性はとても高いと考えられる。