

# Homework-2

---

## 请描述什么事依赖倒置原则，为什么有时候依赖倒置原则又被称为好莱坞原则？

---

依赖倒置原则在GOF中的定义：

1. High-level modules shouldn't depend on low-level modules, both modules should depend on abstractions.
2. Abstractions shouldn't depend on details. Details depend on abstraction

所以依赖倒置原则就是高阶模块不要依赖低阶模块，二者要通过抽象接口来相互依赖；而且抽象接口不要依赖于具体的实现细节，而具体的实现细节要依赖抽象接口。简单来讲就是依赖倒置原则旨在通过抽象高阶模块和低阶模块之间的交互来反转他们之间经典的依赖关系。

好莱坞原则的描述是：

1. Don't Call Us, We'll Call you.

好莱坞原则就是让低阶模块自己注入到高阶模块中，而高阶模块在需要的时候调用低阶模块。所以可以看到好莱坞原则是将控制权交出去，并且解耦了不同模块之间的依赖关系，当这种解耦是通过抽象接口去做的时候，它就是依赖倒置原则。

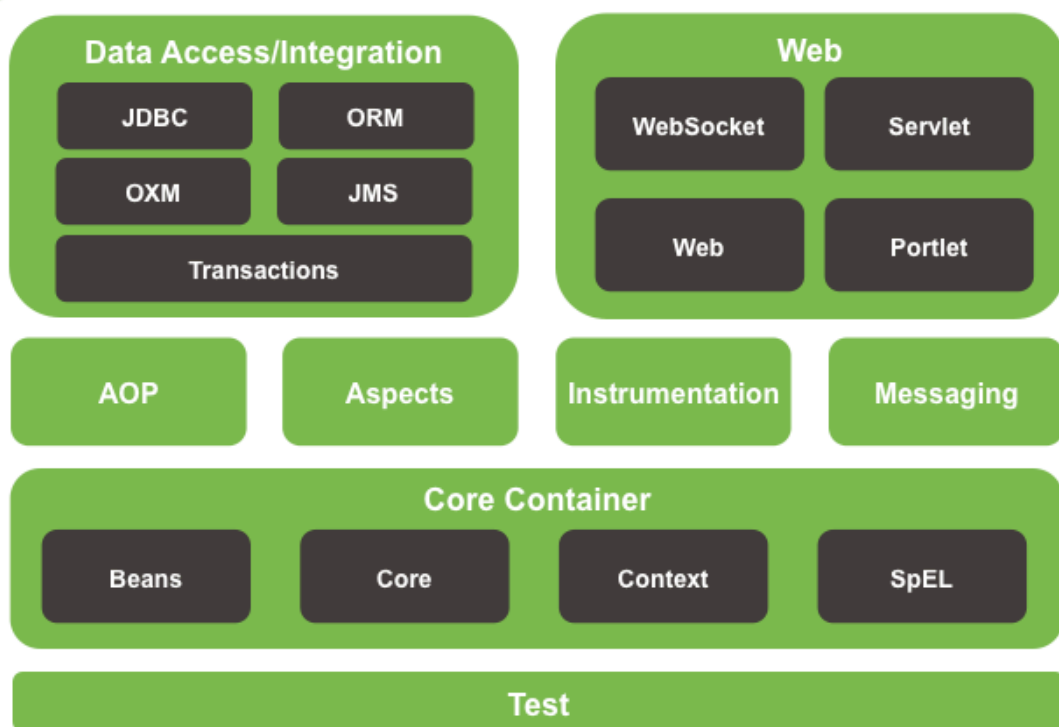
## 请描述一个你熟悉的框架，是如何实现依赖倒置原则的。

---

Spring framework中使用了依赖倒置原则设计整个框架，它的模块和分层介绍图如下，



## Spring Framework Runtime



Spring中的每个模块只负责一个相对独立的功能，模块之间仅有上层对下层的依赖关系，同层之间以及下层对上层几乎没有依赖和耦合，模块之间靠抽象接口耦合，这满足了依赖倒置原则的第一点。

Spring中还对市场流行的主流的中间件，系统访问类库做了进一步的封装和抽象，提供了更高层次，更统一的访问接口，比如，Spring提供了Spring Cache来提供统一的，抽象的缓存访问接口，这些接口不依赖于具体的Cache实现，如Redis, memcache, guava等。在实际使用时，我们可以基于spring提供的统一接口访问Cache，这满足了依赖倒置原则的第二点。

## 参考

1. <https://docs.spring.io/spring/docs/5.0.0.RC2/spring-framework-reference/overview.html>
2. [https://en.wikipedia.org/wiki/Dependency\\_inversion\\_principle](https://en.wikipedia.org/wiki/Dependency_inversion_principle)
3. <https://martinfowler.com/bliki/InversionOfControl.html>