# Spring一些理解

1.Spring有分层的体系结构，这意味着你能选择使用它孤立的任何部分  
2.Spring是一个开放源代码的设计层面框架，他解决的是业务逻辑层和其他各层的松耦合问题  
3.Spring是用于测试驱动工程的理想的framework。  
4.面向切面编程，实现的目的是针对业务处理过程中的切面进行提取，它所面对的是处理过程中的某个步骤或阶段，以获得逻辑过程中各部分之间低耦合性的隔离效果。  
5.Spring包含并管理应用对象的配置和生命周期，在这个意义上它是一种容器  
6.Spring可以将简单的组件配置、组合成为复杂的应用  
7.Spring 客户端发送请求，服务器控制器调用对应的处理器Controller，Controller可以调用一些Service、 DAO来进行数据操作, ModelAndView用于存放从DAO中取出的数据，还可以存放响应视图的一些数据。Spring框架提供一个视图组件ViewResolver，该组件根据Controller返回的标示，找到对应的视图，将响应response返回给用户。  
8.Spring通过一种称作控制反转（IoC）的技术促进了低耦合。当应用了IoC，一个对象依赖的其它对象会通过被动的方式传递进来，而不是这个对象自己创建或者查找依赖对象  
9.Spring的作用是整合，Spring可以将简单的组件配置、组合成为复杂的应用，Bean 工厂  
10.Spring 框架是一个分层架构，由一些定义良好的模块组成  
11.Spring框架的每个模块（或组件）都可以单独存在，或者与其他一个或多个模块联合实现

# 参考学习资料

spring模块：

学习参考：<http://wiki.jikexueyuan.com/project/spring/>

intellij 搭建spring

学习参考：<http://blog.csdn.net/cflys/article/details/70598903>

书：spring 实践第四版

# 学习过程中的问题及笔记

1. 自动化装配：组件扫描

在实现类里增加@Component

在Java配置文件里增加@Configuration、@ComponentScan

单元测试代码增加@RunWith、@ContextConfiguration

1. 运行单元测试失败：[测试类的@RunWith(SpringJUnit4ClassRunner.class)](mailto:测试类的@RunWith(SpringJUnit4ClassRunner.class))

[类写错了@RunWith(BlockJUnit4ClassRunner.class)](mailto:类写错了@RunWith(BlockJUnit4ClassRunner.class))

1. 实现类@Component组件不写名称也能执行
2. 一个接口有多个实现类：@Autowired自动注入的同时需要增加Qualifier注解并指定实现类的组件名

@Qualifier("aotherAddInterfaceImp")

1. Java的config文件路径：config文件@ComponentScan如果不指定包名只扫描与config文件相同包名的组件，可指定一个或者多个包名，目的是可实现config和java文件分别放在不同的目录中
2. rebuid project：修改了文件目录后，项目编译不通过，重新打开项目，
3. DI：@Autowired,通过构造器或者setter方法把别的bean依赖注入到自己的bean中，自己的bean也可以相同的方式注入到其他bean中或者直接写到测试中验证，实现嵌套bean
4. 显示配置bean：单例bean，引用其他bean的bean(DI依赖注入：依赖其他bean生成的bean

方案一：生成自身bean时直接调用已生成的引用类CalculatorInterfaceImp的bean

方案二：生成自身bean时调用通过参数声明引用类传入的bean

方案三：通过setter方法注入其他类

1. XML-配置：

方案一：元素constructor-arg、property(DI用ref属性、设值注入用value属性)

方案二：使用c-/p命名空间：（模式声明后才能使用、复杂的数据类型不可用这种方式，如list，set）

1. c:sub-ref=" " ----sub为构建器参数名
2. c:\_0-ref=" "---0为sub的索引，XML命名不能用数字开头，所以在前面加了“\_”
3. 程序包org.aspectj.lang.annotation不存在

在Project Structure:Libraries:+:From Maven里搜索aspectjrt.jar并增加

# intellij 使用技巧

接口生成实现类：alt+enter

属性上右键：Generate

打印的快捷键：sout