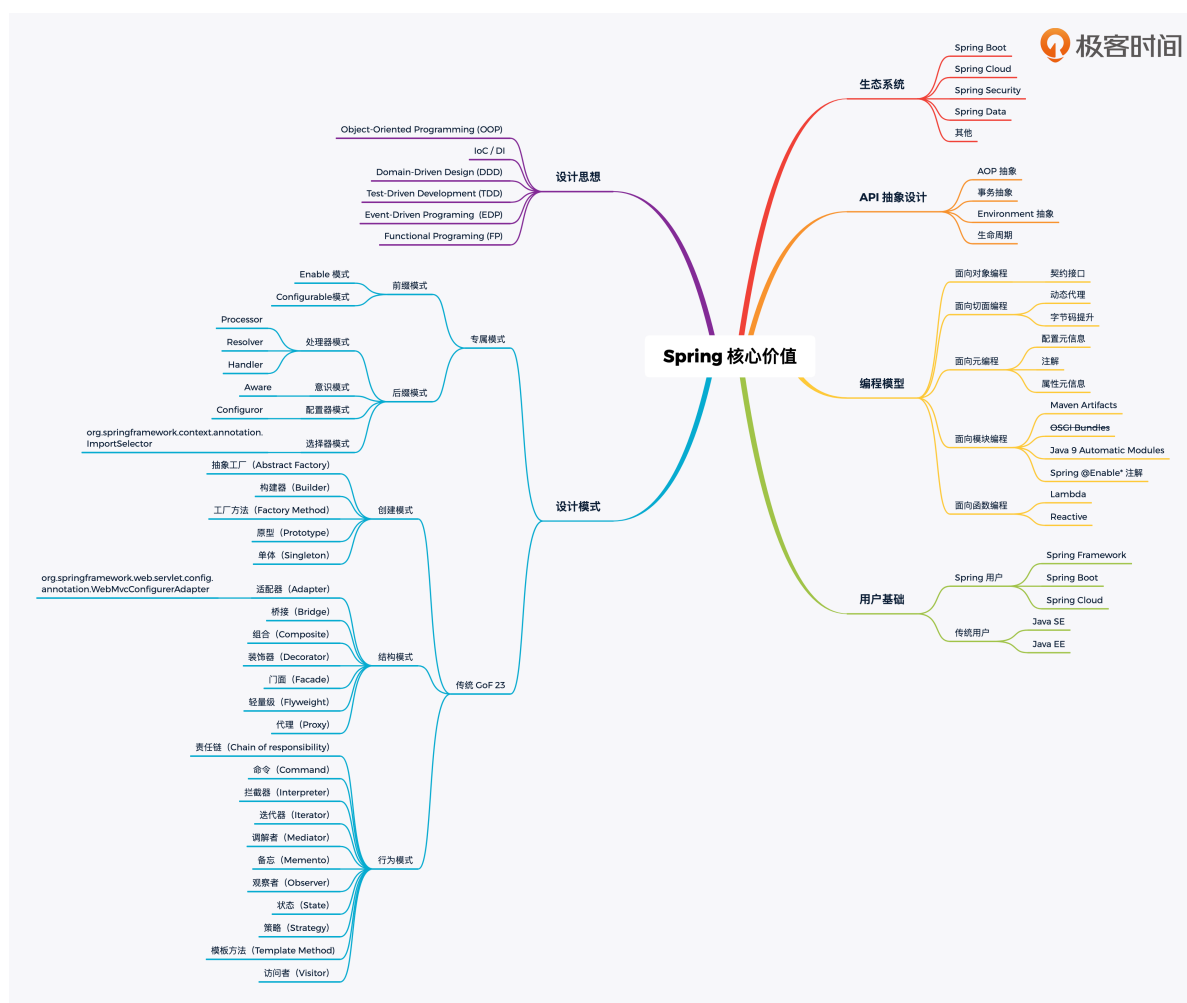
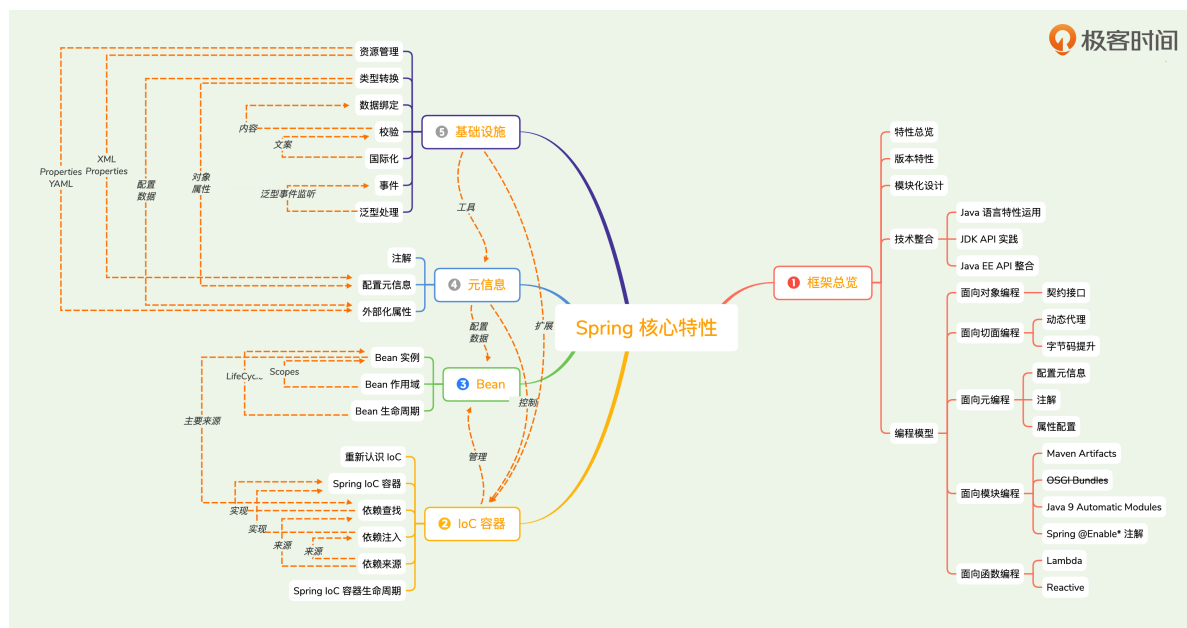


Spring和ORM框架



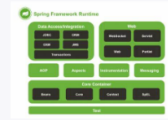
Spring的东西太多，自己总结的没有小马哥好，还是上面把小马哥总结的脑图贴上。Spring Framework是Spring系列技术的基石，Spring Framework里面最为核心的就是IoC 和 Aop。Spring Framework将XML、注解、配置类的对象自动生成Bean，加载到Map，然后通过注入的方式添加各个Bean引用中。这样Spring Framework就可以管理这些Bean。Aop实际上就是通过Jdk的动态代理或者字

节码技术，进行切面增强，用的最多就是数据库事务。

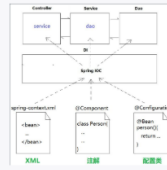
ORM的框架目前主要是Hibernate和MyBatis，Hibernate主要的优点是全自动化的ORM框架，配置好之后，不需要写代码，但是Hibernate转化逻辑比较复杂。相反MyBatis是半自动，需要额外写mapper文件，但是SQL语句是原生，复杂的SQL方便DBA审查。

Spring

Spring Framework



AOP
接口类型 默认使用JdkProxy
非接口类型 默认使用CGLib



DI

- Bean加载过程
- 1.构造函数
 - 2.依赖注入
 - 3.BeanNameAware
 - 4.BeanFactoryAware
 - 5.ApplicationContextAware
 - 6.BeanPostProcessor前置方法
 - 7.InitializingBean
 - 8.自定义init方法
 - 9.BeanPostProcessor后置方法
 - 10.使用中
 - 11.DisposableBean
 - 12.自定义destroy方法



- Spring Bean配置方式演化
- | 版本 | 配置方式 | 注解 |
|----------------|---------------|----------------------------|
| 1.0/2.0 | XML配置/注解注入 | XML/@AutoWire |
| 2.5 | @Service | 半自动注解配置 |
| 3.0 | Java Config配置 | @Bean @Configuration |
| 4.0/SpringBoot | 全自动注解配置 | @Condition @AutoConfigureX |

Spring Boot

- 目的
- 让开发变简单
 - 让配置变简单
 - 让运行变简单
- 特性
- 创建独立运行的Spring应用
 - 直接嵌入Tomcat或Jetty、Undertow，无需部署WAR
 - 提供限定性的starter依赖简化配置
 - 在必要时自动化配置Spring和其他三方依赖库
 - 提供生产production-ready特性，例如指标度量，健康检查，外部配置等
 - 完全零代码生产和不需要XML配置
- 核心
1. 自动化配置：简化配置核心，基于Configuration、EnableXX、Condition
 2. spring-boot-starter：脚手架核心，整合各种第三方类库，协同工具
- 条件化自动配置
- @ConditionalOnBean
 - @ConditionalOnClass
 - @ConditionalOnMissingBean
 - @ConditionalOnProperty
 - @ConditionalOnResource
 - @ConditionalOnSingleCandidate
 - @ConditionalOnWebApplication

ORM

- JPA Java Persistence API
- 一个全自动的ORM框架，将POJO与数据库表建立映射关系，自动生成SQL语句，自动执行
- Hibernate
- 可以使用HQL，Criteria，Native SQL操作数据库
 - 优点 简单场景不用写SQL
 - 缺点 对DBA不友好
- MyBatis
- 半自动化的ORM框架
 - 需要使用映射文件mapper.xml定义map规则和SQL
 - 需要定义mapper/DAO，基于xml规则，操作数据库
 - 优点 原生SQL，直观，对DBA友好
 - 缺点 繁琐
 - Tips 使用工具生成基础的mapper.xml和mapper/dao，其他的类继承生成的mapper

