# 事务属性的种类：   传播行为、隔离级别、只读和事务超时

* 传播行为：**定义了被调用方法的事务边界。**

PROPAGATION\_REQUIRED--支持当前事务，如果当前没有事务，就新建一个事务。这是最常见的选择。

PROPAGATION\_SUPPORTS--支持当前事务，如果当前没有事务，就以非事务方式执行。   
PROPAGATION\_MANDATORY--支持当前事务，如果当前没有事务，就抛出异常。   
PROPAGATION\_REQUIRES\_NEW--新建事务，如果当前存在事务，把当前事务挂起。   
PROPAGATION\_NOT\_SUPPORTED--以非事务方式执行操作，如果当前存在事务，就把当前事务挂起。   
PROPAGATION\_NEVER--以非事务方式执行，如果当前存在事务，则抛出异常。

* 隔离级别：**在 spring 事务中提供了 5 种隔离级别来对应在SQL 中定义的 4 种隔离级别**

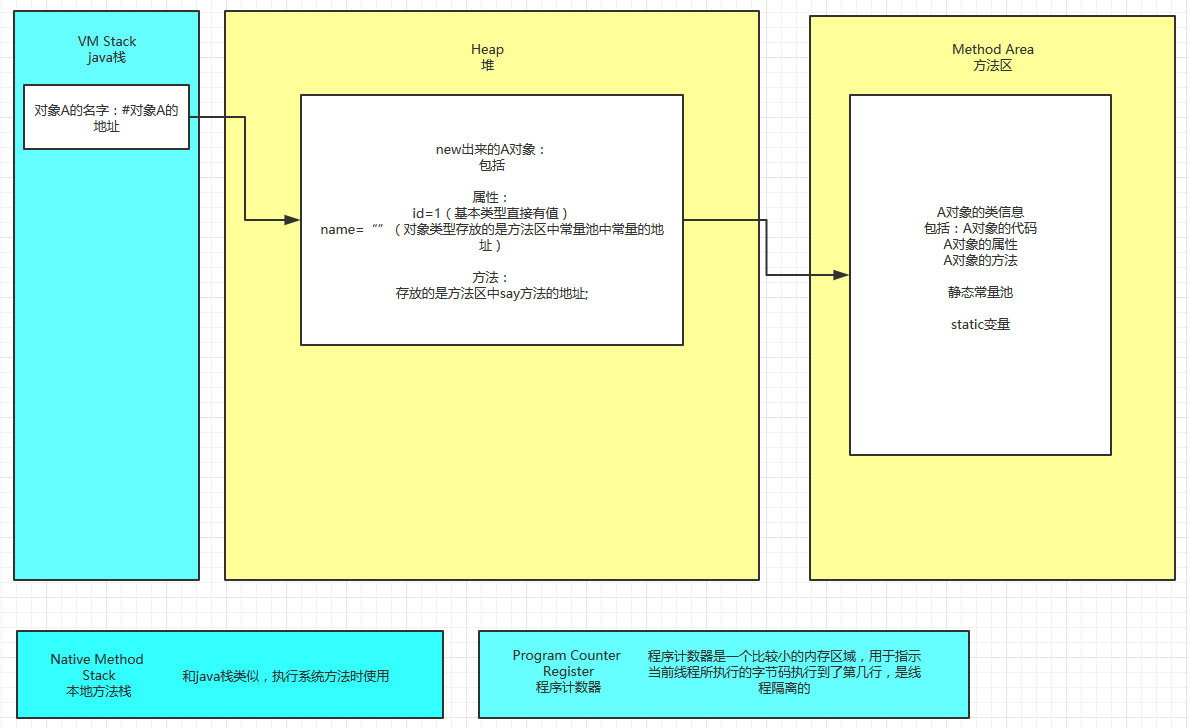
|  |  |
| --- | --- |
| **隔离级别** | **意义** |
| **ISOLATION\_DEFAULT** | **使用后端数据库默认的隔离级别** |
| **ISOLATION\_READ\_UNCOMMITTED** | **允许读取未提交的数据（对应未提交读），可能导致脏读、不可重复读、幻读** |
| **ISOLATION\_READ\_COMMITTED** | **允许在一个事务中读取另一个已经提交的事务中的数据（对应已提交读）。可以避免脏读，但是无法避免不可重复读和幻读** |
| **ISOLATION\_REPEATABLE\_READ** | **一个事务不可能更新由另一个事务修改但尚未提交（回滚）的数据（对应可重复读）。可以避免脏读和不可重复读，但无法避免幻读** |
| **ISOLATION\_SERIALIZABLE** | **这种隔离级别是所有的事务都在一个执行队列中，依次顺序执行，而不是并行（对应可序列化）。可以避免脏读、不可重复读、幻读。但是这种隔离级别效率很低，因此，除非必须，否则不建议使用。** |

* 只读：**如果在一个事务中所有关于数据库的操作都是只读的，也就是说，这些操作只读取数据库中的数据，而并不更新数据，那么应将事务设为只读模式（ READ\_ONLY\_MARKER ） , 这样更有利于数据库进行优化 。**
* 事物超时：**如果一个事务长时间运行，这时为了尽量避免浪费系统资源，应为这个事务设置一个有效时间，使其等待数秒后自动回滚。**

# Servlet的生命周期分为5个阶段：加载、创建、初始化、处理客户请求、卸载。

* (1)加载：容器通过类加载器使用servlet类对应的文件加载servlet
* (2)创建：通过调用servlet构造函数创建一个servlet对象
* (3)初始化：调用init方法初始化
* (4)处理客户请求：每当有一个客户请求，容器会创建一个线程来处理客户请求
* (5)卸载：调用destroy方法让servlet自己释放其占用的资源

# JVM





# 责任链模式

责任链模式（Chain of Responsibility Pattern）为请求创建了一个接收者对象的链。这种模式给予请求的类型，对请求的发送者和接收者进行解耦。这种类型的设计模式属于行为型模式。

在这种模式中，通常每个接收者都包含对另一个接收者的引用。如果一个对象不能处理该请求，那么它会把相同的请求传给下一个接收者，依此类推。

使用场景：如登陆校验与角色校验。Java中的Filter（过滤器）和Interceptor（拦截器）

# HTTPS

https协议需要到ca申请证书，一般免费证书很少，需要交费。  
http是超文本传输协议，信息是明文传输，https 则是具有安全性的ssl加密传输协议 http和https使用的是完全不同的连接方式用的端口也不一样：前者是80，后者是443。  
　　http的连接很简单，是无状态的 HTTPS协议是由SSL+HTTP协议构建的可进行加密传输、身份认证的网络协议 要比http协议安全