Atitit 架构原则api 类库版本兼容 循序渐进 避免革命大摧毁模式

[1.1. 实现类库版本兼容有以下几个目标 1](#_Toc12828)

[1.2. 原则上老人老办法，新人新办法，只增加新的，不修改旧的，除非确定的bug 1](#_Toc4894)

[1.3. 尽可能也使用ioc模式，隔离类库的方法关联，增加build方法貌似不错 1](#_Toc8506)

[1.4. 可以减少外部初始化过程。方便快捷。 1](#_Toc28205)

[1.5. api提供过程。一次提供，终身负责制。 1](#_Toc16752)

[2. 几个api层次， 1](#_Toc31111)

[2.1. 方法函数模式 ，meth\_v2 meth3 这种模式 2](#_Toc23311)

[2.2. 类模式，当方法改动很多时候，可以复制为cls2模式，原来的cls1 保持简单不懂了。 2](#_Toc31424)

[2.3. 命名空间模式 2](#_Toc27070)

[3. 可以参考jdk os win如何实现兼容性 2](#_Toc8639)

## 实现类库版本兼容有以下几个目标

用户可以共同混用高低版本吧的类库jar，，而不会产生问题。。

常常a组件依赖c1类库，而b组件依赖c2类库。。可能混用。。要保证没问题。

源码管理，最好一份即可，，一份包含所有的进化点与版本。。方便管理。

## 原则上老人老办法，新人新办法，只增加新的，不修改旧的，除非确定的bug

## 尽可能也使用ioc模式，隔离类库的方法关联，增加build方法貌似不错

## 可以减少外部初始化过程。方便快捷。

## api提供过程。一次提供，终身负责制。

## 不要修改以前的旧版本api签名

架构原则：adapter，vm模式

# 几个api层次，

## 方法函数模式 ，meth\_v2 meth3 这种模式

例如

## 类模式，当方法改动很多时候，可以复制为cls2模式，原来的cls1 保持简单不懂了。

Cls cls2 clsV2 模式。。

要不要把cls2里面的mth2 改成mth。。我看其实也没有必要。。

## 命名空间模式

Apache的lang3 就是彻底的改变了命名空间。。

# 可以参考jdk os win如何实现兼容性