摘自：http://blog.csdn.net/dreamboy87/article/details/17244467

l.什么是模块化

与面向对象一样，模块化的目的也是松耦合，高内聚。我们可以理解为模块化是将对象间的互访做了边界划分，即对一组业务相关的对象进行封装，并且提供可能的更高层次的代码访问隔离机制。

2.物理模块化VS逻辑模块化

物理模块化是指应用中的类文件被物理的分割放在不同的模块中，但是每个模块间的互访不受控制，各个模块可以访问模块间的内部对象，只要对象是可访问的。只是是对代码本身进行模块化管理。

例如JAVA中，应用被分为模块A和B，模块B中有一个public对象B.b，该对象可以完全被模块A访问，因为它是public的。

逻辑模块化是指在物理模块化的基础上，对模块进行控制访问；即模块间实现了访问隔离，而这才是我们所说的真正的模块化的概念。

再看回上面的例子，如果B.b没有定义是其它模块可访问的，那么默认A.a是访问不到B.b这个对象的，不管这个对象的访问级别是什么。

3.OSGI的作用

在java中，OSGI是一个实现java模块化互访的平台，我们可以理解为是一个更高级的JVM。它提供了逻辑上的模块化控制。

4.OSGI对模块的定义

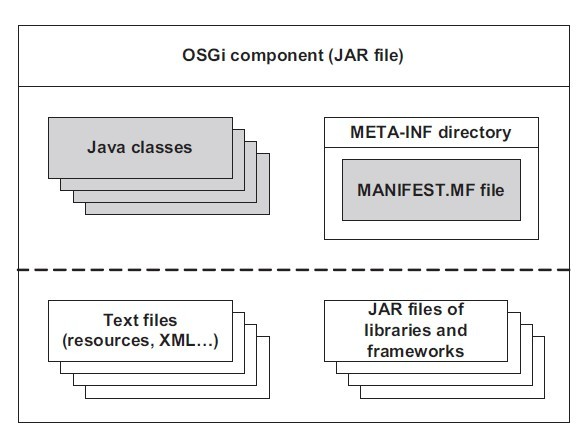
在OSGI中，模块称之为bundle，一个bundle在物理上而言就是一个jar包。Jar包中有一个描述jar包的信息文件，位于jar内部的META-INF目录下的MANIFEST.MF文件。OSGI通过MANIFEST这个文件获取模块的定义信息，比如模块间的互访信息，模块的版本信息等。

5.bundle里有什么

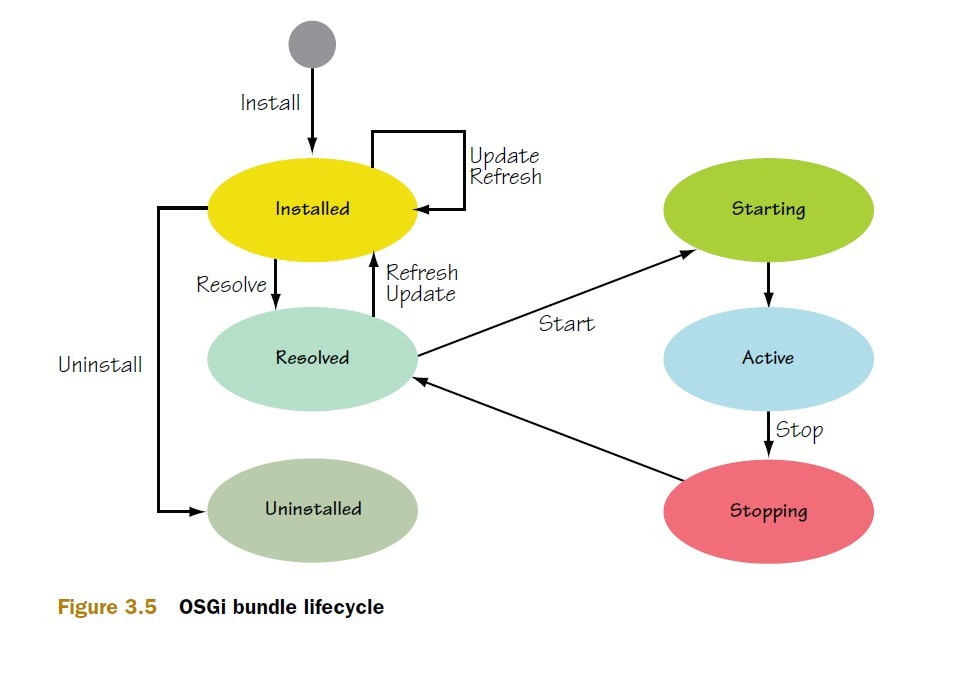
一个bundle中一般包含如下的东西：

部署描述文件（MANIFEST.MF，必要的），各类资源文件（如html、xml等，非必须的），还有类文件。这与一个普通的jar包没有任何的区别。但是，除此之外，bundle里还可以放入其它的jar包，用于提供给bundle内部的类引用，即bundle内部的lib库。（跟一个war很类似）。

Note：实际上bundle里可以存放任何的内容，但是在bundle内部不会有嵌套的bundle，即上面提到的存放于bundle中的jar包就只会当成是一个普通的jar包，不管这些jar包中是否含有bundle定义的信息



6.Bundle的生命周期



如图，椭圆表示状态，bundle在OSGI中一共可以查看到5个状态，分别是Installed，Resolved，starting，Active，stopping。Uninstalled则是一个虚的状态，即当Bundle不存在于OSGI中时的状态。

OSGI里对Bundle的操作一共有6个，分别是Install，Update，Refresh，Start，Stop，Uninstall。

通过这些操作就可以达到不同的状态。

7.什么是OSGI Service？

我们有了export和import，已经可以很好的管理模块间的代码互访，但是，如果我们只希望曝露接口，而不是实现的话，如何让依赖的Bundle获得接口的实现呢？这个时候我们就需要一种类似服务查找的方式通过OSGI来获得实现了。所以Bundle有两种角色：生产者和消费者。

生产者Bundle通过注册服务将实现注入OSGI中，消费者Bundle则是通过查找服务的方式从OSGI中获得实现。

OSGI目前提供了2种方式用于注册和查找服务：

(1)编程式

通过BundleContext进行注册或查找。

(2)声明式

是OSGI R4.0中引入的新的方式，通过XML文件的配置进行服务的注册和查找。