我们通常对同一个系统，需要将各个模块的工程，其中的基础性的、相同的依赖，全部放入一个父工程中，集中式统一约束所有模块的依赖，避免每个模块负责的人胡乱使用依赖版本，比如有的人用spring 3，有的人用spring 4

如果每个模块是不同的人负责的，结果就可能导致一个问题，有可能：

（1）有的人可能就是某一天，突然就想升级spring版本到5.0，结果其他人用的都是4.0，最后导致项目集成的时候就出各种问题

（2）比如项目里现在需要用一个组件，activiti，结果每个模块的负责人各自加入自己的依赖，有的人用1.0，有的人用2.0，有的人用3.0，完全没有统一起来

maven里面提供了一个功能，叫做继承，也就是说，我们可以将一个项目中所有模块通用的一些配置，比如依赖和插件，全部放在一个父工程中，oa-parent，然后所有的子工程，声明一下从父工程中去继承

如果直接将所有的依赖放入父工程，然后子工程用<parent>元素声明继承，然后此时会强制性继承父工程中所有的依赖和插件。

但是实际的场景下，不同的工程需要的依赖是不一样的，可能有的工程需要从父工程继承20个依赖和3个插件；有的工程需要从父工程继承10个依赖和2个插件。如果按照上面那种最low最普通的做法，会导致每个子工程都强制性从父工程继承所有的依赖。

那么这就导致，本来某个子工程只要10个依赖，现在导致每次打包，assembly的大包，可能导致每个工程的jar包打包速度变慢，打完以后的包比较大。这种做法是很不好的，不推荐。

推荐的一个做法，是在父工程中，使用<depdendencyManagement>和<pluginManagement>两个元素，来声明要被子工程继承的依赖和插件

有一个好处，就是，如果子工程用<parent>元素声明继承父工程，此时不会强制性继承所有的依赖和插件，子工程需要同时声明，自己要继承哪些依赖和插件，但是只要声明groupId和artifactId即可，不需要声明version版本号，因为version全部放在父工程中统一声明，强制约束所有子工程的相同依赖的版本要一样。

将oa-organ、oa-auth、oa-flow三个工程的pom.xml中加入以下配置

<parent>

<groupId>com.zhss.oa</groupId>

<artifactId>oa-parent</artifactId>

<version>1.0.0-SNAPSHOT</version>

</parent>

然后将所有工程相同的依赖和插件的配置，全部放入父工程中，此时子工程就会继承父工程的依赖和插件，不需要每个工程都重复定义，而且可以在父工程中统一约束所有依赖和插件的版本。

但是这里有个问题，如果直接将所有依赖和插件直接放入父工程，会导致子工程强制性继承，也许子工程并不需要某些依赖呢？

此时最好的做法，就是在父工程中，使用<dependencyManagement>元素和<pluginManagement>元素，来声明所有的依赖和插件。

此时在子工程中，就可以对自己需要的依赖进行声明，但是不用定义版本号，只要groupId和artifactId即可。这样声明之后才会继承依赖，而且版本由父工程约束。

提一点，这里很关键的一个知识点，是你要搞清楚，dependencyManagement和pluginManagement的意义在哪儿？一般来说，很多人都没搞清楚用上面说的那两个元素来进行继承，以及直接在parent里面放dependencies和plugins里面放依赖和插件来继承，他们俩的区别在哪儿？

如果在父工程中，直接用dependencies和plugins来声明依赖和插件，子工程会强制全部继承；如果用dependencyManagemnent和pluginManagement来声明依赖和插件，默认情况下，子工程什么都不继承，只有当子工程声明了某个依赖或者插件的groupId+artifactId，但是不指定版本时，才会从父工程继承那个依赖。在规范的工程管理中，肯定都是用dependencyManagement和pluginManagemnent的。853769620