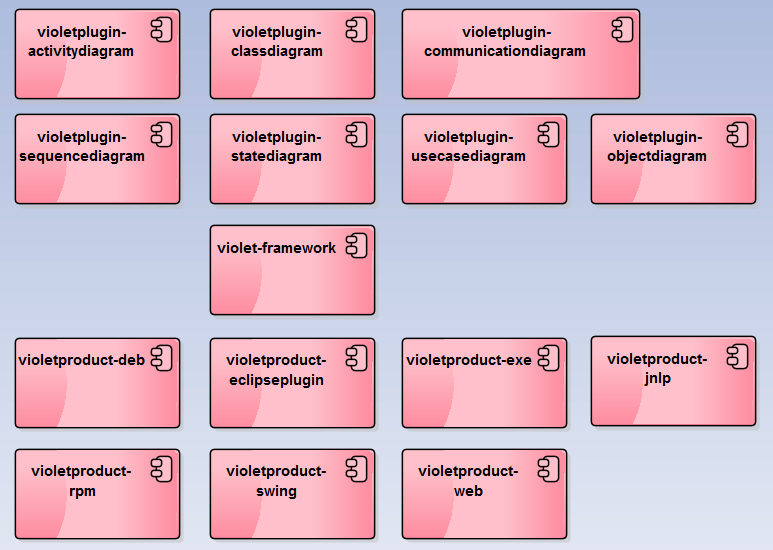
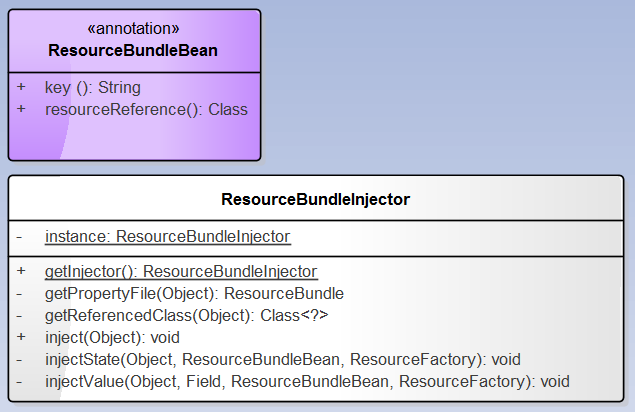
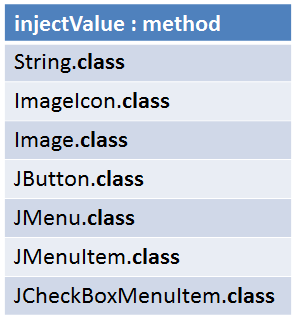
1、入口

Main-Class: com.horstmann.violet.UMLEditorApplication

2、violetumleditor的15个组件：



3、ResourceBundleBean注解

**ResourceBundleBean的作用是查找资源和国际化。**

ResourceBundleInjector是ResourceBundleBean的注解处理器。

ResourceBundleBean既可作用于类也可作用于字段。分别对应于ResourceBundleInjector中的injectState和injectValue方法。其中injectValue稍微复杂一点有对七种情况的处理（如上图），具体情况请参考源码。

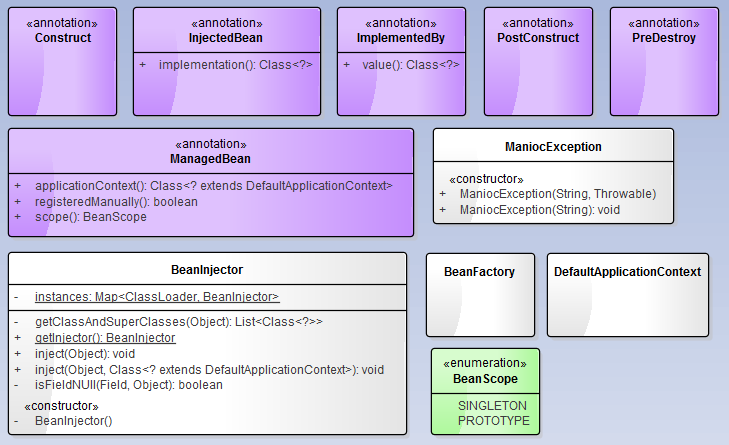
示例：参考SplashScreen类

4、ManiocFramework

ManiocFramework应该是我目前看到的最复杂的类了。

ManiocFramework内部包含6个注解、2个注解处理器（静态内部类）、一个枚举类、一个自定义异常（静态内部类）和一个DefaultApplicationContext（空实现，静态内部类）。

如下图：



BeanFactory的实现非常复杂：



截止2016年4月1日02:00:01，我把所有的方法都看了一遍，大概知道它们提供什么服务。但是，还是需要进一步的理解。待下次再次深入研究。

5、codec

XHTMLPersistenceService类中有使用org.apache.commons.codec.binary.Base64OutputStream

所以特意看了下codec，发现比较难懂，源码中充斥着移位操作符。像这种操作二进制的代码，更偏向于计算机（计算机硬件原理）。所以，理解起来比较难。

如果搜索org.apache.commons.codec可以找到一些使用教程。教程都比较简单，就是教你如何对一个数据进行加密操作。codec有Base64、md5、sha1、sha256Hex等加密方式，Base64应该是唯一既提供加密同时提供解密的API。

6、对violet的研究暂停（2016年4月1日18:18:38）。我研究violet的起因是，好奇其保存工程的方式（保存为一个HTML文件）。图像是保存为data:image/png;base64编码格式的字符串。（PS：为何会突然想起百度的ZRender呢？！）

先这样吧。上传到git！