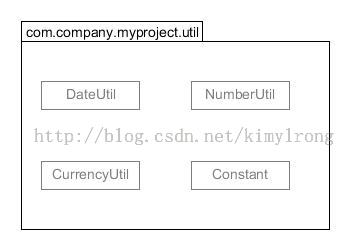
[UML之包图(Package Diagram)](http://blog.csdn.net/kimylrong/article/details/40043389)

软件的架构设计，思路遵循由大及小。相较于Class Diagram，Package Diagram从更宏观的角度来展示软件的架构设计，主要体现在代码组织方面。Package Diagram对一些大型的项目特别有用。顺便说一句，良好的代码组织，对软件的可维护性至关重要，请认真的规划你的包结构。下文将简单介绍Package Diagram，主要分两块，什么是Package，以及Package之间的关系。

## 什么是Package

用最简单的方式来说，Package可以理解为文件夹(folder)。代码的组织从大到小，分为三个层次:文件夹层，文件层，以及文件内部的块(Block)层(函数块之类的)。Package体现的就是文件夹层。Java里面可能是一串文件夹，比如java.lang、java.util等，也叫Package；C++里面，Package对应的是namespace，虽然不能完全等同于文件夹，不过也可以往这边靠；其它的如Node.js，Python等大都体现在文件夹层。

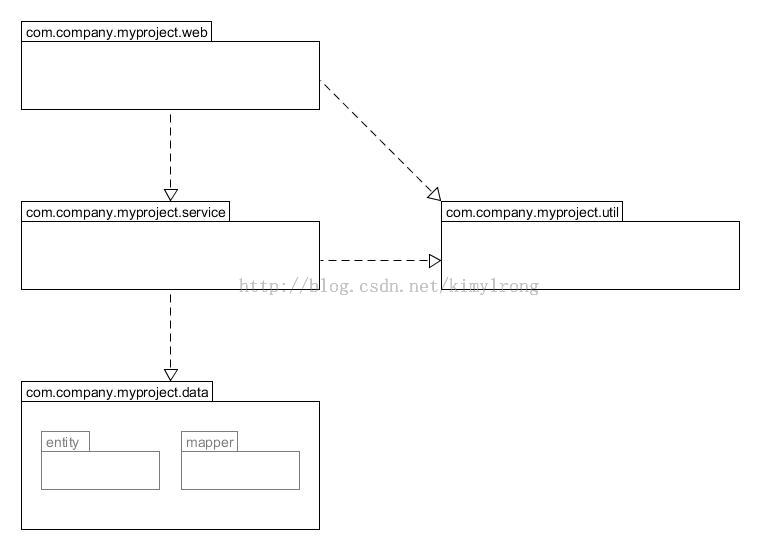
Package在UML里面用一个Tab框表示，Tab里面写上Package的名字，框里面可选地填充一些其它子元素，如类，子Package等。Package的名字可以写全称，也可以简写，风格可可参考项目所用语言的惯例。下面是一个简单的Package示例:



上图为某java项目的一个工具Package，里面有四个工具类。类比较多的时候，为简单明晰起见，也可以不填写子元素。

## Package之间的关系

如果只有一个Package，那也就失去了Package的作用，没有画Package Diagram的意义。Package之间的关系非常的简单，两个字，依赖，UML中依赖用带箭头的虚线表示。我个人还是非常不建议依赖关系出现Cycle的。依赖关系最常见的一个例子就是分层架构，把代码分布到多个层次中，某层可以依赖于下层以及同层，但是不能依赖于上层。其它的组织方式还包括按照模块划分，按照功能划分等。下图是一个Java项目的简单三层架构Package Diagram：



示例中三个主要的Package：data用于访问数据库，也叫dao，Mybatis项目里面分为entity和mapper两个子Package，service是业务逻辑的组件，web用于接受http请求。util为通用组件。

## 小结

Package Diagram从宏观角度展示了项目的组织架构，在大型项目中，是重要的一种UML图。可以先Package Diagram，再Class Diagram的方式来展示项目的架构。