**Factory Method（工厂方法）**

意图

定义一个用于创建对象的接口，让子类决定实例化哪一个类。Factory Method使一个类的实例化延迟到其子类。

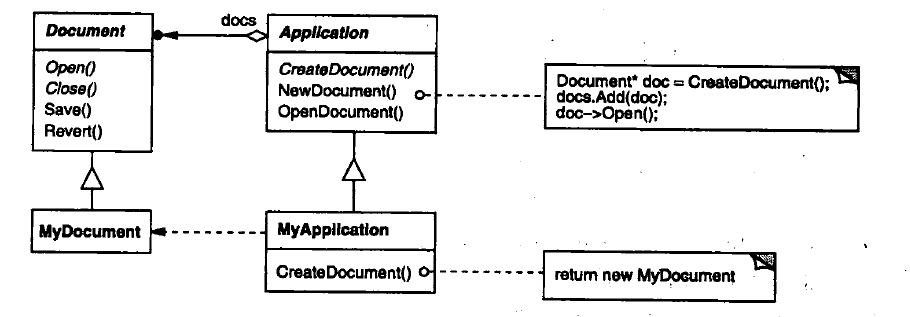
动机

框架使用抽象类定义和维护对象之间的关系。这些对象的创建通常也由框架负责。

考虑这样一个应用框架，它可以向用户展示多个文档。在这个框架中，两个主要的抽象是类Application和Document。这两个类都是抽象的，客户必须通过它们的子类来做与具体应用相关的实现。例如，为创建一个绘图应用，我们定义类DrawingApplication和DrawingDocument。Application类负责管理Document并根据需要创建它们—例如，当用户从菜单中选择Open或New的时候。

因为被实例化的特定Document子类是与特定应用相关的，所以Application类不可能预测到哪个Document子类将被实例化—Application类仅知道一个新的文档何时应被创建，而不知道哪一种Document将被创建。这就产生了一个尴尬的局面：框架必须实例化类，但是他只知道不能被实例化的抽象类。

Factory Method模式提供了一个解决方案。它封装了哪一个Document子类将被创建的信息并将这些信息从该框架中分离出来，如下图所示。



Application的子类重定义Application的抽象操作CreateDocument以返回适当的Document子类对象。一旦一个Application子类实例化以后，它就可以实例化与应用相关的文档，而不需知道这些文档的类。我们称CreateDocument是一个工厂方法，因为它负责生产一个对象。

结构图

