**AbstractFactory（抽象工厂）**

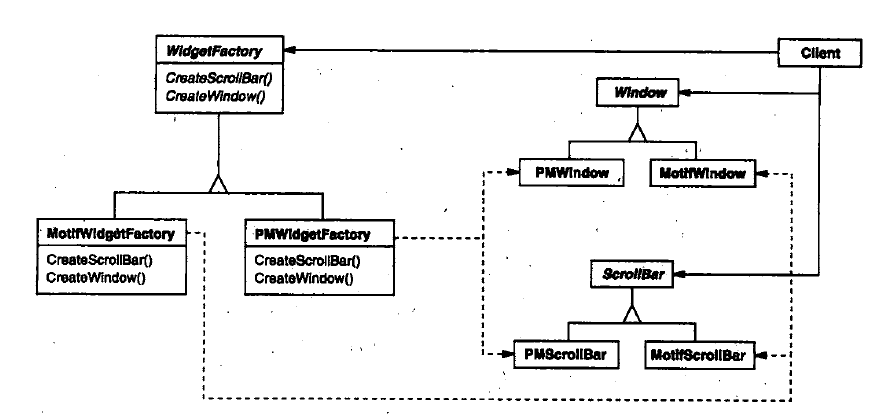
意图

提供一个创建一系列相关或相互依赖对象的接口，而无需指定它们具体的类。

动机

考虑一个支持多种视感标准的用户界面工具包，例如Motif和Presentation Manager。不同的视感风格为诸如滚动条、窗口和按钮等用户界面“窗口组件”定义不同的外观和行为。为保证视感风格标准间的可移植性，一个应用不应该为一个特定的视感外观硬编码它的窗口组件。在整个应用中实例化特定视感风格的窗口组件类将使得以后很难改变视感风格。

为解决这一问题我们可以定义一个抽象的WidgetFactory类，这个类声明了一个用来创建每一个类基本窗口组件的接口。每一类窗口组件都有一个抽象类，而具体子类则实现了窗口组件的特定视感风格。对于每一个抽象窗口组件类，WidgetFactory接口都有一个返回新窗口组件对象的操作。客户调用这些操作以获得窗口组件实例，但客户并不知道他们正在使用的是哪些具体类。这些客户就不依赖于一般的视感风格，如下图所示。



每一种视感标准都对应于一个具体的WidgetFactory子类。每一子类实现那些用于创建合适视感风格的窗口组件的操作。例如，MotifWidgetFactory的CreateScrollBar操作实例化并返回一个Motif滚动条，而相应的PMWidgetFactory操作返回一个Presentation Manager的滚动条。客户仅通过WidgetFactory接口创建窗口组件，他们并不知道哪些类实现了特定视感风格的窗口组件。换言之，客户仅与抽象类定义的接口交互，而不使用特定的具体类的接口。

WidgetFactory也增强了具体窗口组件类之间依赖关系。一个Motif的滚动条应该与Motif按钮、Motif正文编辑器一起使用，这一约束条件作为使用MotifWidgetFactory的结果被自动加上。

结构图

