**Prototype（原型）**

意图

用原型实例指定创建对象的种类，并且通过拷贝这些原型创建新的对象。

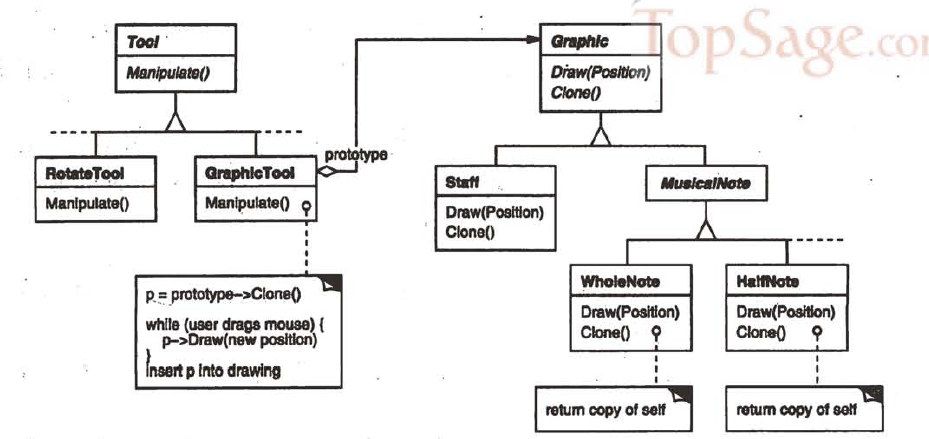
动机

你可以通过指定一个通用的图形编辑器框架和增加一些表示音符，休止符和五线谱的新对象来构造一个乐谱编辑器。这个编辑器框架可能有一个工具选择板用于将这些音乐对象加到乐谱中。这个选择板可能还包括选择、移动和其他操纵音乐对象的工具。用户可以点击四分音符工具并使用它将四分音符加到乐谱中。或者它们可以使用移动工具在五线谱上上下移动一个音符，从而改变它的音调。

我们假定该框架为音符和五线谱这样的图形构件提供了一个抽象的Graphics类。此外，为定义选择面板中的那些工具，还提供一个抽象类Tool。该框架还为一些创建图形对象实例并将它们加入到文档中的工具预定义一个GraphicTool子类。

但GraphicTool给框架设计者带来一个问题。音符和五线谱的类特定于我们的应用，而GraphicTool类却属于框架。GraphicTool不知道如何创建我们的音乐类的实例，并将它们添加到乐谱中。我们可以为每一种音乐对象创建一个GraphicTool的子类，但这样会产生大量的子类，这些子类仅仅在它们所初始化的音乐对象的类别上有所不同。我们知道对象复合是比创建子类更灵活的一种选择。问题是，该框架怎么样用它来参数化GraphicTool的实例，而这些实例由Graphic类所支持创建的。

解决方法是让GraphicTool通过拷贝或者克隆一个Graphic子类的实例来创建新的Graphic，我们称这个实例为一个原型。GraphicTool将它应该克隆和添加到文档中的原型作为参数。如果所有Graphic子类都支持一个Clone操作，那么GraphicTool可以克隆所有种类Graphic，如下图所示。



因此在我们的音乐编辑器中，用于创建音乐对象的每一种工具都是一个用不同原型进行初始化的GraphicTool实例。通过克隆一个音乐对象的原型并将这个克隆添加到乐谱中，每个GraphicTool实例都会产生一个音乐对象。

我们甚至可以进一步使用Prototype模式来减少类的数目。我们使用不同的类来表示全音符和半音符，但可能不需要这么做。它们可以是使得不同位图和时延初始化的相同的类的实例。一个创建全音符的工具就是这样的GraphicTool，它的原型是一个被初始化成全音符的MusicalNote。这可以极大的减少系统中类的数目，同时也更易于在音乐编辑器中增加新的音符。

结构图

