**Composite（组合）**

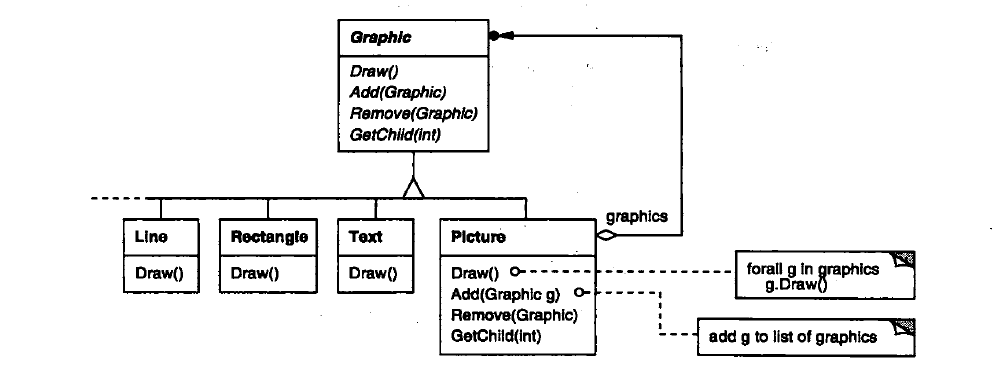
意图

将对象组合成树形结构以表示“部分—整体”的层次结构。Composite使得用户对单个对象和组合对象的使用具有一致性。

动机

在绘图编辑器和图形捕捉系统这样的图形应用程序中，用户可以使用简单的组件创建复杂的图表。用户可以组合多个简单组件以形成一些较大的组件，这些组件又可以组合成更大的组件。一个简单的实现方法是为Text和Line这样的图元定义一些类，另外定义一些类作为这些图元的容器类（Container）。

然而这些方法存在一个问题：使用这些类的代码必须区别对待图元对象与容器对象，而实际上大多数情况下用户认为它们是一样的。对这些类区别使用，使得程序更加复杂。Composite模式描述了如何使用递归组合，使得用户不必对这些类进行区别，如下图所示：

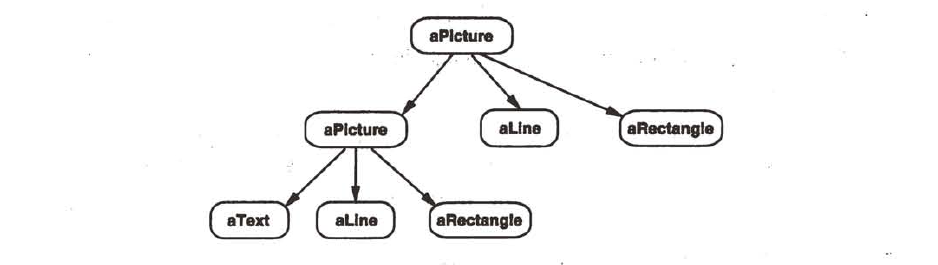


Composite模式的关键是一个抽象类，它既可以代表图元，又可以代表图元的容器。在图形系统中的这个类就是Graphic，它声明一些与特定图形对象相关的操作。例如Draw。同时它也声明了所有的组合对象共享的一些操作，例如一些操作用于访问和管理它的子部件。

子类Line，Rectagle和Text定义了一些图元对象，这些类实现Draw，分别用于绘制直线、矩形和正文。由于图元都没有子图形，因此它们都不执行与子类有关的操作。

Picture类定义了一个Graphic对象的聚合。Picture的Draw操作是通过对它的子部件调用Draw实现的，Picture还用这种方法实现了一些与其子部件相关的操作。由于Picture接口与Graphic接口是一致的，因此Picture对象可以递归地组合其他Picture对象。

下图是一个典型的由递归组合的Graphic对象组成的组合对象结构。



结构图

