1. 如何在Composite模式中避免环状引用？

在Component使用数组（或其他容器）记录引用路径route。根节点的引用路径为空。自根节点开始每次调用add(Component c)增加成员时，记录新成员的引用路径。比如假设有Component c1和c2，调用c1.add(c2)时，检查c1的引用路径中是否有c2，若有，则抛出异常，不要把c2加入到c1的成员中。若没有，则可以加入，并设置c2的引用路径为c1.route.append(c1)。这样杜绝了Component的引用路径上有重复节点的现象，自然也就避免了环状引用。

注意组合模式使对象形成树形结构。由于树中任何节点都不能有多于一个的父节点，则认为任何Component都只能作为成员被添加到其他节点一次，即调用c1.add(c2)时，如果c2的引用路径已经不为空，则直接抛出异常，不必再进行与c1引用路径的对比检查。上述解决方案基于这样的预设。

1. 在对象适配器中，一个适配器能否适配多个适配者？如果能，应该如何实现？如果不能，请说明原因？如果是类适配器呢？

类适配器通过继承Adaptee实现，而对象适配器通过聚合（维持对Adaptee的引用来实现。因此，一个对象适配器只要聚合多个类（同时维持对多个不同Adaptee的引用），就可以适配多个不同的适配者。然而，一个类适配器只能适配一个适配者（类），因为它只能是一个特定Adaptee的子类。

1. JDBC/ODBC桥梁是Bridge模式吗？

桥接模式的定义是将抽象部分与实现部分解耦，使两者都能独立变化。它用抽象关联取代传统的多层继承，类之间的静态继承关系转换为动态组合。

JDBC/ODBC在一定程度上实现了抽象与实现部分的解耦，如针对实现部分有抽象的Driver接口对应Implementor，而MysqlDriver、OracleDriver等具体数据库的Driver对应Concrete Implementor，可对不同类型的数据库进行动态组合。然而，抽象部分只有DriverManager起到扩充抽象类的作用，没有抽象类，即实现部分是可以独立变化的，而抽象部分只是保留了理论上的独立变化的可能，实际上没有体现出变化。可以说是一种简化的Bridge模式。