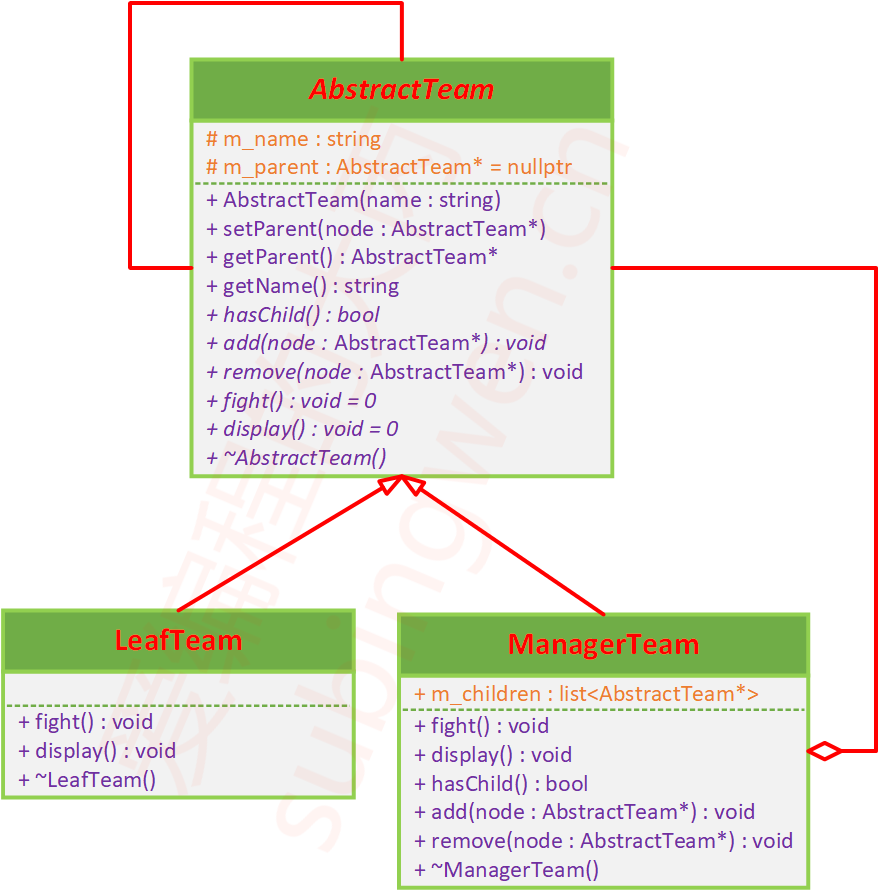
接下来我们进入**组合模式**。

给出**定义**：组合多个对象形成树形结构以表示”部分-整体”的结构层次。组合模式对单个对象(即叶子对象)和组合对象(即树这只鸟容器对象)的使用具有一致性。组合模式又可以称为”部分-整体”模式，属于对象的结构模式，将对象组织到树形结构中去，可以用来描述整体与部分间的关系。



我们先来分析一个组合模式下的UML类图，其实都很简单。在接下来中，我们将会使用树的术语进行分析。

其实组合模式抽象理解起来很简单，就是将一系列具有相似属性的类使用树的数据结构给它管理起来。在这种设计下，我们的UML类图中存在一个抽象基类，这个基类其实就是我们建树中常用的节点类。在这个抽象基类中，包含了一系列类似于树的方法以及在实际中我们这个类所扩展出来的属性。

在这些方法中，为了实现组合模式，其中的关键就是那个自身抽象类的指针。这个指针的作用我们也很熟悉，就是我们建树中常用的那个边。这里我们一般不考虑使用左右孩子的俩个指针设计，毕竟这样更耗费空间，也可能导致我们后续在设计时出现混乱。所以我们使用父亲节点这个在多叉树中每个节点唯一的属性进行连接。

在这个抽象基类的简单派生中，我们可以看到它派生出了俩个子类，其中一个是一个叶子节点类，并没有什么特殊的。另一个是管理节点类。好，这里我们对整颗组合模式将要组成的树进行下分析和分类。在这一整颗树中，我们将最底层的节点(即没有子节点的节点)分类为叶子节点。在叶子节点之上的，统一分类为中间节点，或者说管理节点(因为这些个节点在逻辑上都有着自己的孩子)。在这些个管理节点中，除没有父亲的根节点外，其他的都大致一样。

总的来说，组合模式就是一颗专门用于管理类的树。在这个模式中，每个在这颗树中的节点都有着指针指向它的父亲(当然可能存在设计使得使用孩子指针，不给不太可能就是了)。在这种层次设计下，使得同一层次下的节点往往具有相同或相似的属性，而位于不同分支下的节点，由于深度的加深，各自的属性可能会逐渐的发生差异化，但是总的来说还是会符合整颗树的设计规范的。而且，只要在这颗树中，无论你是位于哪里，只要你遵守了抽象层，也就是抽象节点的接口设计规范，那么所有的节点都可以为外部使用统一的接口进行调用。

对了，既然是一颗树，那么对于这颗树的有效遍历，将会是这颗树无法逃脱的命题，而一个节点值保留其的父节点地址显然是无法实现遍历的。所以这个就需要我们去进行一些额外的设计，在我们的管理节点类中，我们需要去添加一个额外的成员变量来储存我们当前节点所连接的所有的子节点的地址。通过这个成员变量，我们才能够从整颗树的根节点去不断的向下遍历，知道遍历完整颗树。