



# COMPOSITE

- **Operations**
  - `addChild`, `removeChild`, `getChild`, `Iterator`
- **Issues**
  - Explicit parent reference in child; leads to integrity constraint
  - Graph instead of tree; multiple parents
  - How much default behavior to put into `Component`
    - Too much supports `Leaf` but makes for clumsy code
    - One approach is to have the default be "no children" for `Composite`
  - How to track children: vector, list, array, hash table
  - Child ordering
  - `Composite` can cache traversal information on children

```
abstract class Component {  
    abstract int size();  
    protected String display() {}  
}
```

```
class Composite extends Component {  
    int size() {}  
    Component get(int i) {}  
    void add(Component c) {}  
    void add(int index, Component c) {}  
    void removeRange(int start, int end) {}  
    void removeElementAt(int index) {}  
}
```

```
class Leaf extends Component {  
    int size() {}  
}
```

```
class Holophraser {  
    private String display(String t) {}  
    void collapse(int start, int end, String s) {}  
    void expand(int index) {}  
}
```