

1. Composite

The text is store in a composite tree structure, with a Column Glyph as the root, Row Glyphs as the recursive composite, and the characters are stored a Character Glyphs as the leaves.

2. Builder

The GlyphBuilder class acts as a builder, which can build each of the 3 types of Glyphs.

3. Singleton

The GlyphBuilder is implemented as a singleton by having a private constructor, a private static variable called instance, and a public static method called getInstance, which will create a builder if one doesn't already exist, and then return instance.

4. Flyweight

The flyweight pattern is implemented within the GlyphFactory. The factory has a Character pool that is used when creating Character objects with a certain char value.

5. Bridge

The GoFList is implemented using an ArrayList, which is how the bridge pattern is implemented.

6. Adapter

GoFListAdapter adapts the Java ArrayList to the desired GoFList interface.

7. Abstract Factory

The class GlyphFactory creates the 3 types of nodes to be used in the composite structure.

8. Iterator

The pre order implementation of an Iterator is used here, which a common interface called GlyphIterator, with a NullIterator, PreorderIterator, and a ListIterator being its children. The iterator is used to traverse the tree, and combined with the Visitor pattern, will extract the text from it.

9. Visitor

The DisplayTextVisitor is used to extract each character from the leaf nodes, and return the completed string.

10. Command

The command pattern is implemented by having the façade create an UpdateCommand object to perform a task. Having this command objects makes it easier to perform the undo and redo operations, with the aid of the memento pattern of course.

11. Memento

The memento pattern is used to save the state of the tree whenever a command is performed. This makes it very easy to implement undoing and redoing actions.

12. Façade

The SystemFaçade class acts as an intermediary between the UI and the underlying subsystem. This protects the subsystem by only allowing 1 point of contact between the two sections.