

Taller 5 – Patrones de Diseño

DISEÑO Y PROGRAMACIÓN ORIENTADA A OBJETOS
DANNA LUCIA CASTILLO PALACIOS

Tabla de contenido

Librería JGraphX	1
Información general del proyecto.....	2
Propósito y funcionalidad	3
Patrón Compuesto (Composite)	4
Escribir el título del capítulo (nivel 2).....	5
Escribir el título del capítulo (nivel 3).....	6
Implementación del Patrón compuesto en JGraphX	4
Escribir el título del capítulo (nivel 2).....	5

Librería JGraphX

1.0 Información general del proyecto

1.0.1 Propósito y funcionalidad

JGraphX se utiliza para construir y representar visualmente grafos en aplicaciones Java. Proporciona un conjunto completo de funcionalidades para crear, editar y visualizar grafos, así como interactuar con ellos. JGraphX puede utilizarse en una amplia gama de aplicaciones que involucren representaciones gráficas de relaciones y estructuras complejas.

JGraphX admite una amplia gama de funciones para permitir que la visualización de celdas solo esté limitada por la habilidad del desarrollador y la plataforma Swing. Los vértices pueden ser formas, imágenes, dibujos vectoriales, animaciones, prácticamente cualquier operación gráfica disponible en Swing. También puede usar marcado HTML en etiquetas de texto. (JGraph, 2020)



1.1 Estructura general del diseño

1.1.1 Generalidades del diseño

El diseño de JGraphX sigue una estructura modular y jerárquica. La biblioteca se compone de varios componentes clave, como mxGraph, mxGraphModel, mxCell, mxGeometry, mxStyleSheet y mxGraphView, que interactúan entre sí para proporcionar la funcionalidad completa. Estos componentes se basan en principios de diseño orientado a objetos y patrones de diseño para facilitar la extensibilidad y la personalización.

1.1.2 Retos de diseño

- Representación eficiente de grandes grafos: JGraphX debe ser capaz de manejar eficientemente grafos grandes con miles o incluso millones de elementos.
- Rendimiento y optimización: La representación y manipulación de gráficos pueden ser operaciones intensivas en términos de rendimiento. JGraphX debe optimizar su implementación para proporcionar una experiencia fluida incluso con grandes gráficos y operaciones complejas.
- Coherencia visual y estilos: Mantener la coherencia visual de los elementos del grafo y aplicar estilos consistentes en toda la representación gráfica puede ser un desafío, especialmente cuando se trata de elementos interconectados.

1.1.3 URL del Proyecto

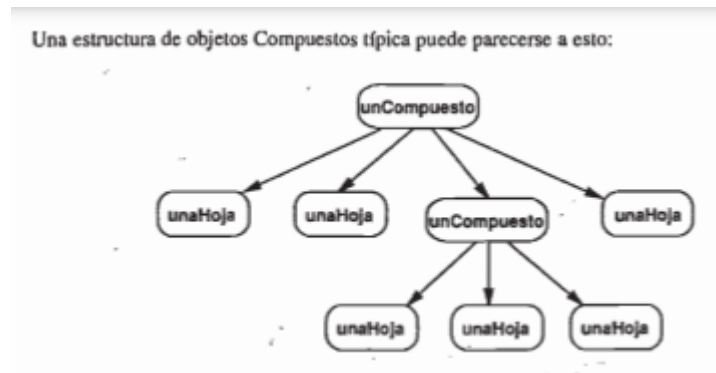
<https://github.com/jgraph/jgraphx.git>

Patrón Compuesto (Composite)

1.0 El patrón Compuesto

El patrón de diseño Composite es un patrón estructural que permite tratar a los objetos individuales y a las composiciones de objetos de manera uniforme. Proporciona una estructura jerárquica en forma de árbol, donde los nodos hoja y los nodos compuestos se tratan de la misma manera.

En el patrón Composite, se define una interfaz común llamada "Componente" que representa tanto a los objetos individuales como a las composiciones de objetos. Los objetos individuales se denominan "hojas" y los objetos compuestos se denominan "compuestos". Los compuestos contienen una lista de componentes hijos, que pueden ser tanto hojas como otros compuestos.



1.1 Usos frecuentes del patrón Compuesto

El patrón Composite se utiliza cuando se desea tratar los objetos individuales y las estructuras de objetos compuestas de manera uniforme. Algunos casos de uso comunes del patrón Composite son:

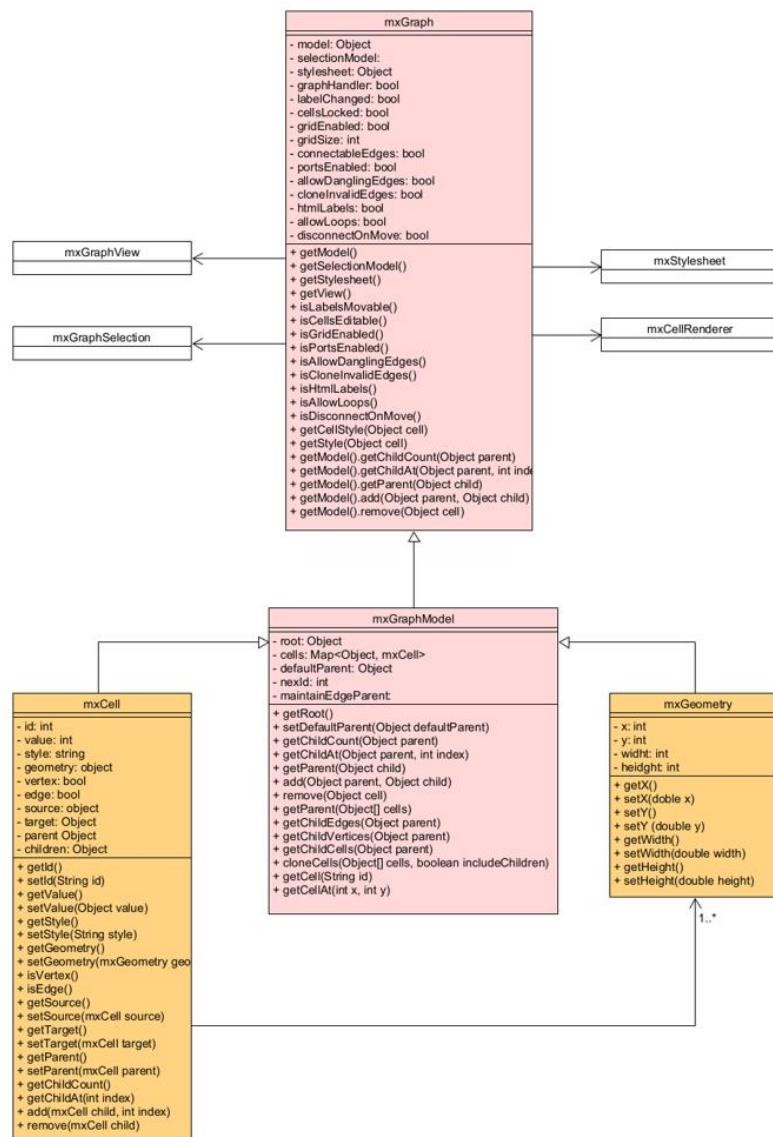
- Representación de estructuras jerárquicas: El patrón Composite permite representar estructuras jerárquicas complejas mediante la composición recursiva de objetos.

- Tratamiento uniforme de elementos individuales y compuestos: El patrón Composite permite tratar tanto los elementos individuales como los compuestos de la misma manera, lo que simplifica el código y mejora la flexibilidad.
- Recursividad y composición de comportamiento: El patrón Composite permite aplicar operaciones recursivas en la estructura jerárquica de objetos, como realizar un cálculo o aplicar una operación a todos los elementos.
- Manipulación eficiente de estructuras complejas: El patrón Composite facilita la manipulación de estructuras complejas mediante el acceso y la modificación uniforme de los elementos individuales y las composiciones de elementos.

Implementación del Patrón compuesto en JGraphX

1.0 Diagrama de clases

En el diagrama de clases se puede observar de forma general como está implementado el patrón Compuesto en la librería JGraphX



Referencias

JGraph. (28 de Octubre de 2020). *JGraphX (JGraph 6) User Manual*. Obtenido de https://jgraph.github.io/mxgraph/docs/manual_javavis.html#2.2.1