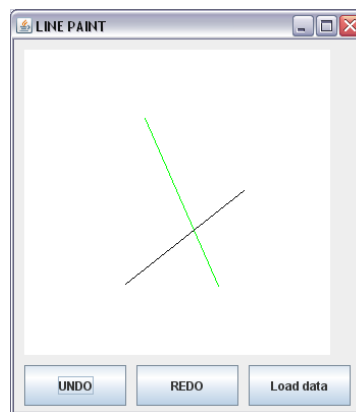


Escuela Colombiana de Ingeniería

Herramienta de dibujo con persistencia on-line.

Se le ha encargado hacer un trabajo de *refactoring* a una aplicación de dibujo, que por ahora, se limita a la diagramación a partir de líneas rectas (por ahora las funciones de UNDO y REDO están deshabilitadas):



El modelo sobre el que funciona esta herramienta, se basa en el metapatrón MVC (modelo-vista-controlador), siendo el siguiente su diagrama de clases:

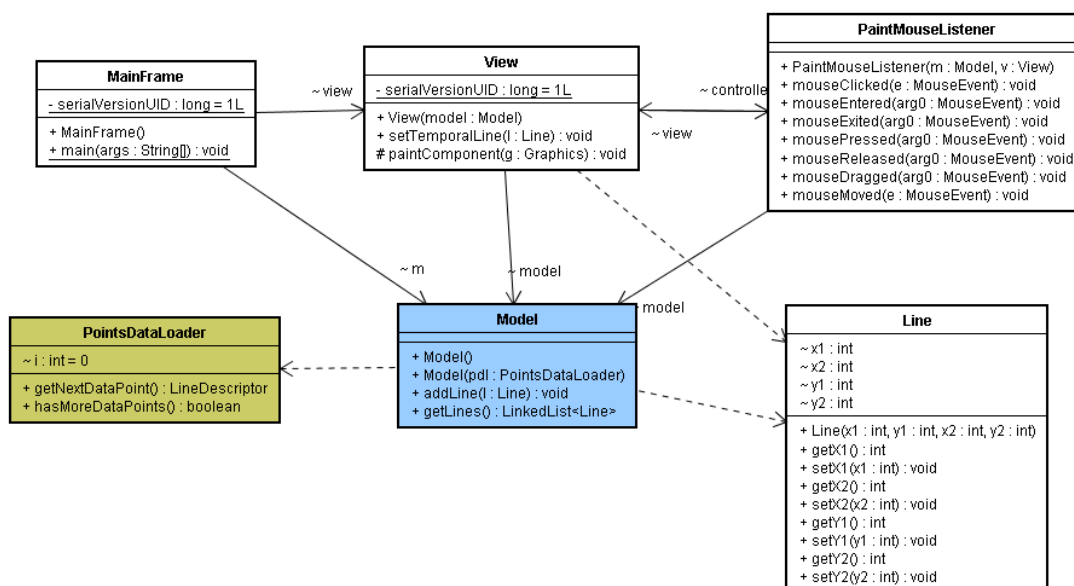


Figura 1 - Diagrama de clases aplicación (MVC)

Teniendo en cuenta el diagrama anterior, la herramienta funciona de la siguiente manera:

1. El frame principal tiene una vista (un componente swing personalizado), un modelo (el cual guarda los vectores que va trazando el dibujante), y un controlador (MouseListener, que se encarga de detectar los eventos del usuario en el área de dibujo y coordinar a la vista y al modelo).
2. Cuando el controlador detecta que el usuario ha realizado un trazo sobre el área de dibujo, agrega al modelo la información del vector, y posteriormente le indica a la vista que debe actualizarse. La vista, grafica el estado actual del modelo.
3. Cuando se selecciona la opción 'Load Data', se crea un nuevo modelo, usando el constructor que recibe un objeto de tipo PointsDataLoader (el cual carga información con coordenadas de recta de un archivo predeterminado dentro del mismo modelo), y crea una nueva vista para el nuevo modelo.

Se puede observar que en la versión actual de la aplicación, al seleccionar la opción "Load Data" se produce un error (analice la traza de la excepción generada para identificar cual es el problema). Se sabe que la aplicación realiza una serie de consultas SQL a una base de datos externa (de otro sistema de información), y que en dicha base de datos, sin previo aviso, se hicieron cambios en los nombres de las tablas. Al revisar un poco el nuevo esquema de base de datos, se encontró que la tabla donde están los datos que se deben graficar en la herramienta ahora se llama DATOSGRAFICADOR.

Su tarea es volver a habilitar la funcionalidad de la aplicación (es decir, que cargue datos al momento de hacer clic en el botón en cuestión), sin dejar de utilizar el código existente en las librerías (ya que resulta demasiado costoso en tiempo re-implementarlas). Identifique y aplique un patrón de diseño estructural que permita de alguna manera 'interceptar' la sentencia incorrecta y reemplazarla por la correcta, de manera que la opción de carga funcione correctamente.

Para lo anterior tenga en cuenta que se conoce que la clase PointDataLoader usa el API de JDBC de la forma tradicional:

```
...  
Statement stm=conn.createStatement()  
stm.executeQuery("...");  
...
```

Igualmente, tenga en cuenta la documentación de las interfaces de Connection y Statement:

<http://docs.oracle.com/javase/7/docs/api/java/sql/Connection.html>

<http://docs.oracle.com/javase/7/docs/api/java/sql/Statement.html>