Documentacion Package Studio y sus Clases

La clase studiosBean es una clase de Java que representa un bean (objeto contenedor) para estudios. La clase implementa la interfaz "studioInterface" y tiene una serie de atributos y métodos para interactuar con una base de datos de estudios.

Atributos:

- **name**: representa el nombre del estudio
- **headQuarters**: representa la sede del estudio
- **numberWorkers**: representa la cantidad de trabajadores en el estudio
- **dateCreation**: representa la fecha de creación del estudio
- **conn**: representa la conexión a la base de datos
- **statement**: representa la declaración para ejecutar consultas en la base de datos

Métodos:

- **loadJDBC()**: carga el controlador JDBC necesario para conectarse a la base de datos
- **connect()**: establece una conexión a la base de datos
- `executeQuery(Connection conn, String query )`: ejecuta una consulta en la base de datos y retorna un objeto ResultSet con los resultados.
- **displayAllStudios(Connection conn, DefaultTableModel model)**: muestra todos los estudios en una tabla especificada por el parámetro model.
- **getName()**: retorna el nombre del estudio
- **setName(String name)**: establece el nombre del estudio
- **getHeadQuarters()**: retorna la sede del estudio
- **setHeadQuarters(String headQuarters)**: establece la sede del estudio
- **getNumberWorkers()**: retorna la cantidad de trabajadores en el estudio
- **setNumberWorkers(int numberWorkers)**: establece la cantidad de trabajadores en el estudio
- **getDateCreation()**: retorna la fecha de creación del estudio
- **setDateCreation(String dateCreation)**: establece la fecha de creación del estudio
- **getConn()**: retorna la conexión a la base de datos
- **setConn(Connection conn)**: establece la conexión a la base de datos
- **getStatement()**: retorna la declaración para ejecutar consultas en la base de datos

- **setStatement(Statement statement)**: establece la declaración para ejecutar consultas en la base de datos

Además, se pueden crear métodos para agregar, actualizar o eliminar estudios en la base de datos utilizando las funciones correspondientes de SQL.

En resumen, la clase **studiosBean** proporciona una forma fácil de interactuar con una base de datos de estudios, permitiendo la ejecución de consultas y la visualización de los datos en una tabla. También permite la creación de nuevos estudios, la actualización de información existente y la eliminación de estudios.

La clase studioInterface es una interfaz que define los métodos que se deben implementar en la clase studiosBean. Estos métodos incluyen:

1. loadJDBC() - Carga el controlador JDBC necesario para conectarse a la base de datos.
2. connect() - Establece una conexión con la base de datos.
3. executeQuery(Connection conn, String query) - Ejecuta una consulta en la base de datos.
4. displayAllStudios(Connection conn, DefaultTableModel model) - Muestra todos los datos de la tabla studios.
5. getName(), setName(String name) - Métodos getter y setter para el atributo nombre.
6. getHeadQuarters(), setHeadQuarters(String headQuarters) - Métodos getter y setter para el atributo headQuarters.
7. getNumberWorkers(), setNumberWorkers(int numberWorkers) - Métodos getter y setter para el atributo numberWorkers.
8. getDateCreation(), setDateCreation(String dateCreation) - Métodos getter y setter para el atributo dateCreation.
9. getConn(), setConn(Connection conn) - Métodos getter y setter para el atributo conn.
10. getStatement(), setStatement(Statement statement) - Métodos getter y setter para el atributo statement.
11. addStudio(Connection conn) - Añade un nuevo estudio en la tabla studios en la base de datos.
12. updateStudio(Connection connection) - Actualiza un estudio existente en la tabla studios en la base de datos.
13. deleteStudio(Connection conn) - Elimina un estudio existente en la tabla studios en la base de datos.

    La clase studiosBean debe implementar estos métodos para tener acceso a la base de datos y realizar operaciones CRUD (crear, leer, actualizar, eliminar) en la tabla studios.

La clase studioGallery es una clase de utilidad que proporciona una fábrica para crear una instancia de la interfaz **studioInterface**. La clase contiene un método estático llamado **getStudioDao()** que devuelve una nueva instancia de la clase **studiosBean**.

La función de esta clase es proporcionar un mecanismo para obtener una instancia de una clase que implementa la interfaz **studioInterface** sin necesidad de conocer la clase específica que se está utilizando. Esto proporciona una mayor flexibilidad en el diseño de la aplicación, ya que se puede cambiar la implementación de la interfaz sin afectar al resto del código.

En resumen, la clase **studioGallery** proporciona una fábrica para crear instancias de la interfaz **studioInterface** mediante el método estático **getStudioDao()**. Esto proporciona una mayor flexibilidad en el diseño de la aplicación al permitir cambios en la implementación de la interfaz sin afectar al resto del código.

La clase main es la clase principal del programa, que contiene el método **main** de ejecución. Crea una instancia de **studioInterface** utilizando la clase **studioGallery** y carga el controlador JDBC, establece una conexión con la base de datos. El programa crea una interfaz gráfica de usuario con una serie de botones para interactuar con la base de datos. Los botones son: "Display Studios", "Add Studio", "Update Studio" y "Delete Studio". Cada botón tiene un ActionListener agregado que ejecuta la acción correspondiente en la base de datos y actualiza la tabla mostrada en la interfaz gráfica de usuario cuando se presiona.

La clase main contiene el código para la interfaz gráfica de usuario (GUI) del programa. Utiliza la clase JFrame para crear una ventana principal que contiene un panel donde se muestran las opciones del menú: "Display Studios", "Add Studio", "Update Studio" y "Delete Studio". Cada una de estas opciones tiene un ActionListener asociado que se activa cuando el usuario hace clic en el botón correspondiente. Por ejemplo, cuando el usuario hace clic en el botón "Display Studios", se crea una nueva ventana que contiene una tabla JTable, que es llenada con los datos de la tabla "studios" en la base de datos mediante el método displayAllStudios en la clase studiosBean. Asimismo, cuando el usuario hace clic en "Add Studio", se llama al método addStudio en la clase studiosBean, que permite agregar un estudio nuevo a la tabla "studios" en la base de datos. Similarmente, cuando el usuario hace clic en "Update Studio", se llama al método updateStudio en la clase studiosBean, que permite actualizar la información de un estudio específico en la tabla "studios" en la base de datos. Por último, cuando el usuario hace clic en "Delete Studio", se llama al método deleteStudio en la clase studiosBean, que permite eliminar un estudio específico de la tabla "studios" en la base de datos. En resumen, la clase main es responsable de crear la interfaz gráfica de usuario del programa y de manejar los eventos de los botones del menú para interactuar con la base de datos mediante la clase studiosBean.