Clase MainWindow:

Este código es la clase principal llamada "MainWindow" que extiende JFrame y permite visualizar diferentes ventanas según la selección del usuario en el menú. La clase contiene una barra de menú que contiene diferentes elementos de menú que representan diferentes ventanas (Empleados, Empresas, Fabricantes, Tienda, Producto, Inventario).

Cada elemento de menú tiene un "ActionListener" asociado que se activa cuando el usuario selecciona un elemento de menú. El "ActionListener" crea una nueva ventana y la agrega al marco principal. Si hay una ventana actual en el marco, se elimina para permitir la visualización de la nueva ventana.

En resumen, esta clase es la encargada de mostrar diferentes ventanas en una aplicación gráfica de usuario utilizando Java y la librería Swing.

Imports:

La clase MainWindow importa las siguientes clases y paquetes:

empresa.EmpresaWindow: clase de ventana para la opción "Empresas" en el menú.

fabricante.FabricantaWindow: clase de ventana para la opción "Fabricantes" en el menú.

inventario.InventarioWindow: clase de ventana para la opción "Inventario" en el menú.

producto.ProductoWindow: clase de ventana para la opción "Producto" en el menú.

tienda.TiendaWindow: clase de ventana para la opción "Tienda" en el menú.

java.awt.BorderLayout: permite el uso de un layout Border para la disposición de los componentes en la ventana.

java.awt.event.ActionEvent: permite el uso de eventos de acción para los elementos del menú.

java.awt.event.ActionListener: permite agregar un oyente de acción para los elementos del menú.

javax.swing.JFrame: permite el uso de una ventana JFrame para la aplicación.

javax.swing.JMenu: permite crear un menú en la ventana.

javax.swing.JMenuBar: permite crear una barra de menú en la ventana.

javax.swing.JMenuItem: permite crear elementos de menú en la barra de menú.

javax.swing.JPanel: permite el uso de paneles para agregar componentes en la ventana.

Constructor:

El constructor de la clase MainWindow establece las siguientes características para la ventana:

Título: "Main Class"

Tamaño: 1024 x 720 pixels

Acción de cierre: cierre de la aplicación cuando se cierra la ventana.

Ubicación: centrada en la pantalla.

Barra de menú: se agrega una barra de menú a la ventana.

Menú:

El menú en la barra de menú tiene una opción llamada "Menú".

Cada opción en el menú, como "Empleados", "Empresas", etc., muestra una nueva ventana cuando se selecciona.

Antes de mostrar una nueva ventana, se verifica si hay un panel actual en la ventana, y si es así, se elimina para que se pueda mostrar la nueva ventana.

La clase MainWindow proporciona una interfaz gráfica para el usuario para interactuar con la aplicación, ofreciendo diferentes opciones para diferentes ventanas y funcionalidades en la aplicación. La clase es esencial para el funcionamiento de la aplicación y se utiliza como punto de partida para todas las operaciones en la aplicación.

Clase EmpleadoWindow

Esta clase se encuentra en el paquete empleado y representa una ventana para visualizar y manipular los datos de los empleados de una empresa. Se implementa utilizando el framework de ventanas de Java Swing y se compone de los siguientes elementos:

Una tabla (JTable) para mostrar los datos de los empleados

Botones (JButton) para agregar, modificar, buscar, eliminar y mostrar todos los empleados.

La clase hereda de JPanel y se utiliza un Layout de tipo FlowLayout para organizar los elementos en la ventana.

La clase EmpleadoWindow utiliza la clase EmpleadoBean como intermediario para acceder a los datos de los empleados, esta clase a su vez se conecta a la clase EmpleadoInterface que es la encargada de las operaciones CRUD y la clase EmpleadoGallery que actua como un administrador de la clase EmpleadoBean.

Para cargar y conectar a la base de datos se invoca a los métodos loadJDBC y connect de la clase EmpleadoBean

Los botones tienen asociados un ActionListener para ejecutar una acción cuando son presionados, los siguientes son los comportamientos de cada botón:

1. Botón "Agregar Empleado": invoca el método addEmpleado de la clase EmpleadoBean para agregar un nuevo empleado a la base de datos.
2. Botón "Modificar Empleado": invoca el método updateEmpleado de la clase EmpleadoBean para actualizar los datos de un empleado seleccionado en la tabla.
3. Botón "Buscar Empleado": invoca el método findEmpleadoById de la clase EmpleadoBean para buscar un empleado por su identificador.
4. Botón "Eliminar Empleado": invoca el método deleteEmpleado de la clase EmpleadoBean para eliminar un empleado seleccionado en la tabla.
5. Botón "Mostrar Todos": invoca el método displayAllEmployees de la clase EmpleadoBean para mostrar todos los empleados en la tabla.

Nota: La logica de las otras clases windows (EmpresaWindow, ProductoWindow, TiendaWindow, FabricanteWindow e InventarioWindow) es igual a la logica de la clase EmpleadoWindow, con excepción de los nombres de los métodos y de las columnas de la tabla.

Atributos:

serialVersionUID: un número único utilizado para la serialización de objetos de esta clase.

empleadoBean: una referencia a la clase EmpleadoBean que contiene la lógica de negocios para realizar diferentes acciones con la información de los empleados.

Constructor: este método inicializa el diseño de la ventana y los componentes gráficos. Carga la conexión JDBC y se conecta a la base de datos. Crea una tabla con los campos relevantes para la información de los empleados. Agrega diferentes botones para realizar acciones con la información de los empleados, como agregar, modificar, buscar y eliminar empleados.

actionPerformed: este método es llamado cuando se presiona uno de los botones y realiza la acción correspondiente a ese botón, llamando a los métodos correspondientes en la clase EmpleadoBean.

Clase EmpleadoBean

Esta clase es responsable de la lógica de negocios para realizar diferentes acciones con la información de los empleados, como agregar, modificar, buscar y eliminar empleados. Utiliza la clase EmpleadoInterface para interactuar con la base de datos y realizar operaciones CRUD.

Métodos:

loadJDBC: este método carga el controlador JDBC necesario para conectarse a la base de datos.

connect: este método establece la conexión a la base de datos.

addEmpleado: este método agrega un nuevo empleado a la base de datos.

updateEmpleado: este método actualiza la información de un empleado existente en la base de datos.

findEmpleadoById: este método busca un empleado en la base de datos por su identificador.

deleteEmpleado: este método elimina un empleado existente en la base de datos.

displayAllEmployees: este método muestra todos los empleados existentes en la base de datos en la tabla de la ventana.

Clase Empleado Interface

La clase java mencionada es una interfaz llamada "EmpleadoInterface". Esta interfaz contiene los siguientes métodos:

loadJDBC: este método carga una biblioteca JDBC necesaria para conectar con una base de datos.

connect: este método se utiliza para conectar con una base de datos.

addEmpleado: este método agrega un empleado a la base de datos

updateEmpleado: este método actualiza la información de un empleado en la base de datos, donde "model" es el modelo de la tabla y "selectedRow" es la fila seleccionada.

findEmpleadoById: este método busca un empleado en la base de datos por su ID, donde "model" es el modelo de la tabla.

deleteEmpleado: este método elimina un empleado de la base de datos, donde "model" es el modelo de la tabla y "selectedRow" es la fila seleccionada.

displayAllEmployees: este método muestra todos los empleados en la base de datos, donde "model" es el modelo de la tabla.

Clase EmpleadoGallery

La clase "empleadoGallery" es una clase que implementa la interfaz "EmpleadoInterface". Esta clase contiene un único método estático llamado "getEmpleadoDao". Este método devuelve una nueva instancia de la clase "EmpleadoBean", que implementa la interfaz "EmpleadoInterface".

La clase "empleadoGallery" se utiliza como un "mediador" o "puente" entre la interfaz "EmpleadoInterface" y la clase "EmpleadoBean". De esta manera, se puede garantizar que todas las clases que implementen la interfaz "EmpleadoInterface" tengan acceso a la misma funcionalidad y métodos.