Nombre del Autor:

Éver José Palomo Mardini

Universidad:

Universidad de Cartagena

Materia:

Programación Orientada a Objetos (POO)

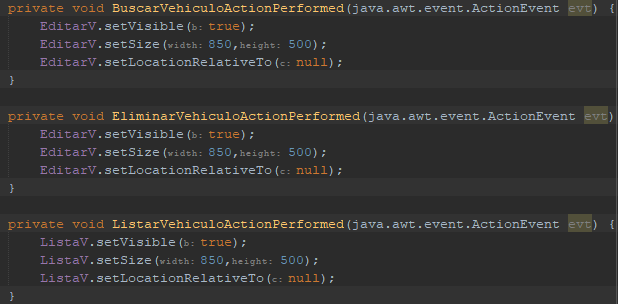
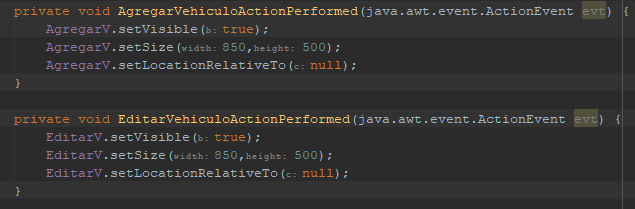
Profesor:

Jhon Arrieta

2023

UNIDAD3

Para esta unidad vamos a utilizar de ejemplo el código de nuestro apartado de Vehículo como base para la explicación del mismo, esto con el fin de que sea más fácil la explicación y no sea tan agobiante mente larga la lectura.

  
  
AgregarVehiculoActionPerformed:

Este método se ejecuta cuando ocurre un evento en el botón "Agregar Vehículo".

Hace que la ventana AgregarV sea visible.

Establece el tamaño de la ventana en 850x500 píxeles.

Coloca la ventana en el centro de la pantalla (setLocationRelativeTo(null)).

EditarVehiculoActionPerformed:

Este método se ejecuta cuando ocurre un evento en el botón "Editar Vehículo".

Hace que la ventana EditarV sea visible.

Establece el tamaño de la ventana en 850x500 píxeles.

Coloca la ventana en el centro de la pantalla.

BuscarVehiculoActionPerformed:

Este método se ejecuta cuando ocurre un evento en el botón "Buscar Vehículo".

Realiza la misma operación que el método EditarVehiculoActionPerformed, haciendo visible la ventana EditarV, ajustando su tamaño y posición.

EliminarVehiculoActionPerformed:

Este método se ejecuta cuando ocurre un evento en el botón "Eliminar Vehículo".

Realiza la misma operación que los métodos anteriores, haciendo visible la ventana EditarV, ajustando su tamaño y posición.

ListarVehiculoActionPerformed:

Este método se ejecuta cuando ocurre un evento en el botón "Listar Vehículos".

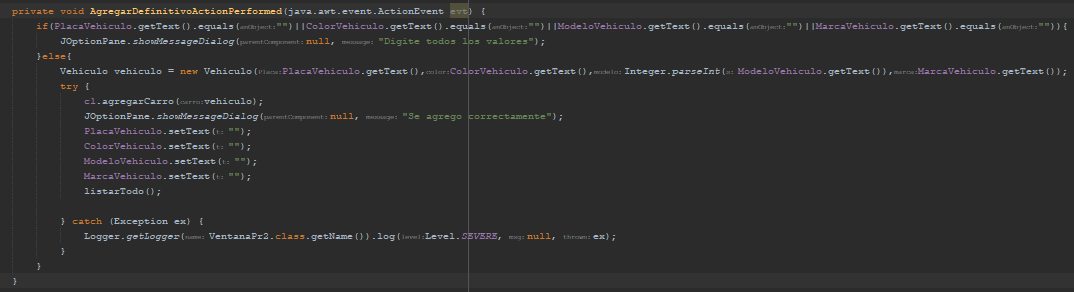
Hace que la ventana ListaV sea visible.

Establece el tamaño de la ventana en 850x500 píxeles.

Coloca la ventana en el centro de la pantalla.

Condición de Verificación:

if(PlacaVehiculo.getText().equals("")||ColorVehiculo.getText().equals("")||ModeloVehiculo.getText().equals("")||MarcaVehiculo.getText().equals("")):: Verifica si alguno de los campos de texto (PlacaVehiculo, ColorVehiculo, ModeloVehiculo, MarcaVehiculo) está vacío. Si al menos uno de ellos está vacío, muestra un mensaje de advertencia.



Bloque de Código en la Condición Verdadera (Campos Vacíos):

JOptionPane.showMessageDialog(null, "Digite todos los valores");: Muestra un cuadro de diálogo (mensaje) informando al usuario que debe completar todos los campos.

Bloque de Código en la Condición Falsa (Campos no Vacíos):

Vehiculo vehiculo = new Vehiculo(PlacaVehiculo.getText(),ColorVehiculo.getText(),Integer.parseInt(ModeloVehiculo.getText()),MarcaVehiculo.getText());: Crea una instancia de la clase Vehiculo utilizando los valores ingresados en los campos de texto.

c1.agregarCarro(vehiculo);: Llama al método agregarCarro de un objeto c1 (presumiblemente de una clase que gestiona vehículos) para agregar el nuevo vehículo.

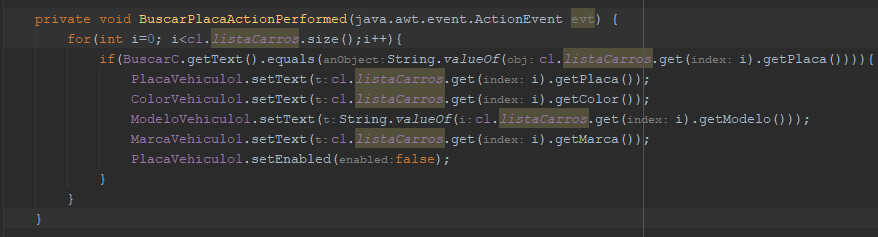
JOptionPane.showMessageDialog(null, "Se agrego correctamente");: Muestra un cuadro de diálogo informando al usuario que el vehículo se agregó correctamente.

Limpia los campos de texto (PlacaVehiculo.setText(""); ColorVehiculo.setText(""); ModeloVehiculo.setText(""); MarcaVehiculo.setText("");) para permitir la entrada de nuevos valores.

listarTodo();: Llama a un método listarTodo (presumiblemente definido en la clase que contiene este código) para actualizar la lista de vehículos en la interfaz gráfica.

Manejo de Excepciones:

catch (Exception ex) { Logger.getLogger(VentanaPr2.class.getName()).log(Level.SEVERE, null, ex); }: Captura y registra cualquier excepción que pueda ocurrir durante la ejecución del bloque de código principal.



Bucle de Búsqueda en la Lista de Carros:

for(int i=0; i<c1.listaCarros.size();i++){: Inicia un bucle for que recorre cada elemento en la lista de carros (c1.listaCarros).

Condición de Búsqueda:

if(BuscarC.getText().equals(String.valueOf(c1.listaCarros.get(i).getPlaca()))){: Compara el contenido del campo de texto BuscarC con la placa del carro en la posición i de la lista. Si encuentra una coincidencia, ejecuta el bloque de código dentro de esta condición.

Bloque de Código en la Condición Verdadera (Placa Encontrada):

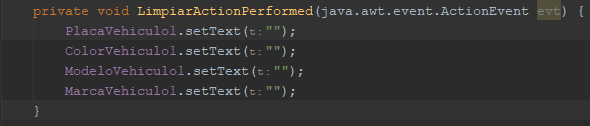
PlacaVehiculo1.setText(c1.listaCarros.get(i).getPlaca());: Establece el texto del campo de texto PlacaVehiculo1 con la placa del carro encontrado.

ColorVehiculo1.setText(c1.listaCarros.get(i).getColor());: Establece el texto del campo de texto ColorVehiculo1 con el color del carro encontrado.

ModeloVehiculo1.setText(String.valueOf(c1.listaCarros.get(i).getModelo()));: Establece el texto del campo de texto ModeloVehiculo1 con el modelo del carro encontrado (convertido a cadena).

MarcaVehiculo1.setText(c1.listaCarros.get(i).getMarca());: Establece el texto del campo de texto MarcaVehiculo1 con la marca del carro encontrado.

PlacaVehiculo1.setEnabled(false);: Desactiva la edición del campo de texto PlacaVehiculo1 para evitar que el usuario lo modifique.



Bloque de Código:

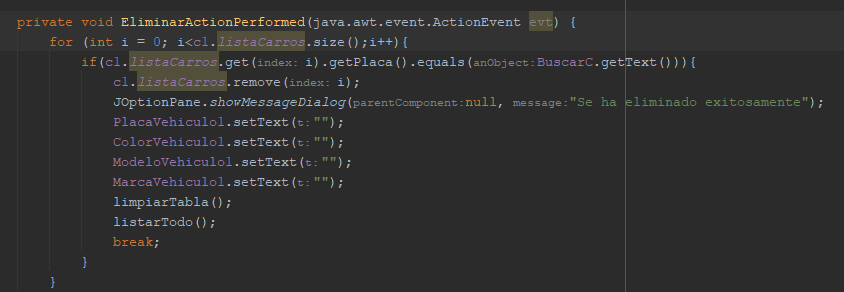
PlacaVehiculo1.setText("");: Establece el texto del campo de texto PlacaVehiculo1 como una cadena vacía, eliminando cualquier texto existente.

ColorVehiculo1.setText("");: Establece el texto del campo de texto ColorVehiculo1 como una cadena vacía.

ModeloVehiculo1.setText("");: Establece el texto del campo de texto ModeloVehiculo1 como una cadena vacía.

MarcaVehiculo1.setText("");: Establece el texto del campo de texto MarcaVehiculo1 como una cadena vacía.

En resumen, este código se encarga de limpiar o borrar el contenido de varios campos de texto en la interfaz gráfica (PlacaVehiculo1, ColorVehiculo1, ModeloVehiculo1, MarcaVehiculo1). Cuando el usuario hace clic en el botón asociado a este código, se eliminan todos los datos presentes en estos campos, lo que proporciona una forma rápida de restablecer o limpiar la información mostrada en la interfaz gráfica.



Bucle de Búsqueda en la Lista de Carros:

for (int i = 0; i<c1.listaCarros.size();i++){: Inicia un bucle for que recorre cada elemento en la lista de carros (c1.listaCarros).

Condición de Búsqueda:

if(c1.listaCarros.get(i).getPlaca().equals(BuscarC.getText())){: Compara la placa del carro en la posición i de la lista con el texto ingresado en el campo de texto BuscarC. Si encuentra una coincidencia, ejecuta el bloque de código dentro de esta condición.

Bloque de Código en la Condición Verdadera (Carro Encontrado):

c1.listaCarros.remove(i);: Elimina el carro de la lista en la posición i.

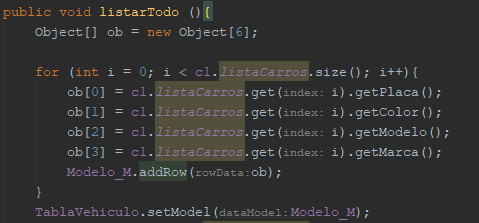
JOptionPane.showMessageDialog(null, "Se ha eliminado exitosamente");: Muestra un cuadro de diálogo informando al usuario que el carro se eliminó con éxito.

Limpia los campos de texto (PlacaVehiculo1.setText(""); ColorVehiculo1.setText(""); ModeloVehiculo1.setText(""); MarcaVehiculo1.setText("");) para reflejar que no hay carro seleccionado.

limpiarTabla();: Llama a un método limpiarTabla (presumiblemente definido en la clase que contiene este código) para limpiar cualquier información visual adicional en la interfaz gráfica.

listarTodo();: Llama a un método listarTodo (presumiblemente definido en la clase que contiene este código) para actualizar la lista de carros en la interfaz gráfica.

break;: Sale del bucle después de eliminar el primer carro coincidente. Esto asume que no hay duplicados de placas en la lista.



Método listarTodo:

Object[] ob = new Object[6];: Declara un array de objetos con tamaño 6. Es probable que esta línea esté destinada a contener los datos de cada carro.

for (int i = 0; i < c1.listaCarros.size(); i++){: Inicia un bucle for que recorre cada elemento en la lista de carros (c1.listaCarros).

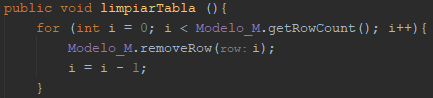
ob[0] = c1.listaCarros.get(i).getPlaca();: Asigna la placa del carro en la posición i al primer elemento del array ob.

ob[1] = c1.listaCarros.get(i).getColor();: Asigna el color del carro en la posición i al segundo elemento del array ob.

ob[2] = c1.listaCarros.get(i).getModelo();: Asigna el modelo del carro en la posición i al tercer elemento del array ob.

ob[3] = c1.listaCarros.get(i).getMarca();: Asigna la marca del carro en la posición i al cuarto elemento del array ob.

Modelo\_M.addRow(ob);: Agrega una nueva fila a un modelo de tabla (Modelo\_M) con los datos contenidos en el array ob. Este modelo de tabla probablemente esté asociado a una tabla en la interfaz gráfica.



Método limpiarTabla:

for (int i = 0; i < Modelo\_M.getRowCount(); i++){: Inicia un bucle for que recorre todas las filas en el modelo de tabla (Modelo\_M).

Modelo\_M.removeRow(i);: Elimina la fila en la posición i del modelo de tabla. Esto limpia la tabla visualmente.

i = i - 1;: Ajusta el índice para compensar la eliminación de la fila.