

TAREA 6



Programación

I.E.S. BRIANDA DE MENDOZA

- CFGS DESARROLLO DE APLICACIONES MULTIPLATAFORMA-

Alumno

Álvaro Bajo Tabero | abajot01@educastillalamancha.es

Profesor

José Miguel Alonso Martínez | jjam53@educastillalamancha.es

7 Enero de 2024

Observaciones de la tarea

El proyecto se podrá encontrar en [GITHUB](#)

Resumen

Paquetes:

El proyecto está dividido en varios paquetes, lo cual es una buena práctica para organizar y estructurar tu código. Algunos de los paquetes son:

- *com.alvarobajo.tare6.data*: Maneja la capa de datos, como el Concesionario y DatosPrueba.
- *com.alvarobajo.tare6.model*: Contiene clases de modelos, como Propietario y Vehiculo.
- *com.alvarobajo.tare6.util*: Contiene clases utilitarias, como UtilComprobaciones y UtilDatos.

Funcionalidades en Paquetes:

- Cada paquete contiene clases relacionadas entre sí en función de su propósito. Por ejemplo, las clases en el paquete data se centran en la manipulación de datos, mientras que las clases en útil se centran en proporcionar utilidades y funciones auxiliares.

Separación de Responsabilidades:

- He intentado hacer una separación clara de responsabilidades entre las clases. Por ejemplo, UtilComprobaciones se encarga de la validación de datos, UtilDatos de la entrada y salida de datos, y Concesionario de la gestión de vehículos.

Entrada/Salida:

- La clase UtilDatos se encarga de la entrada/salida de datos desde y hacia el usuario. Esto es una buena práctica, ya que separa las preocupaciones relacionadas con la interacción del usuario.

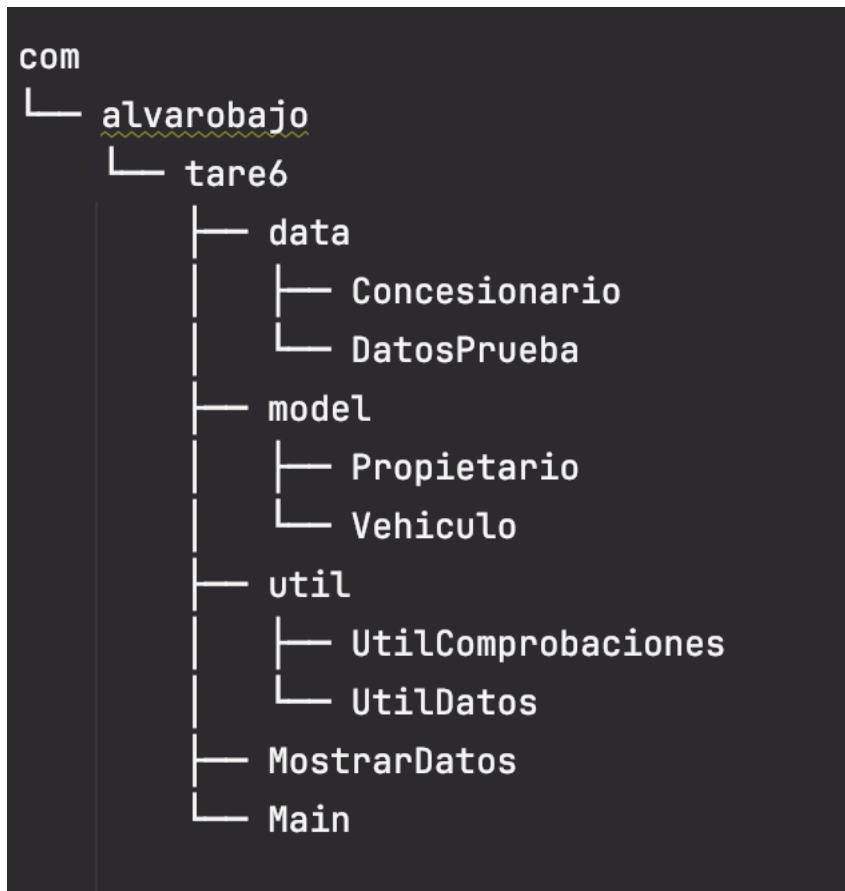
Main Class:

- La clase Main actúa como el punto de entrada principal y contiene el método main. Aquí es donde se inicializa el concesionario, se cargan los vehículos de prueba y se presenta un menú interactivo al usuario.

MostrarDatos:

- La clase MostrarDatos es responsable de mostrar información relacionada con los vehículos en el concesionario.

Representación grafica del proyecto



Observaciones

En este proyecto de gestión de vehículos en un concesionario, he organizado mi código en varias clases y métodos para mejorar la estructura y modularidad del programa. A continuación, proporcionaré un breve resumen de los métodos y sus propósitos, así como las razones detrás de sus implementaciones:

1. **Clase DatosPrueba:**
 - **Propósito:** Contiene datos de prueba para cargar en el concesionario al inicio del programa.
 - **Razón de Uso:** Facilita el desarrollo y las pruebas al tener datos predefinidos.
2. **Clase Concesionario:**
 - **Métodos utilizados:**
 - `insertarVehiculo(Vehiculo vehiculo)`: Inserta un vehículo en el concesionario.
 - `matriculaExiste(String matricula)`: Verifica si una matrícula ya existe en el concesionario.
 - `borrarVehiculo(String matricula)`: Borra un vehículo según su matrícula.

- obtenerListaVehiculos(): Obtiene la lista de vehículos en el concesionario.
 - buscarVehiculo(String matricula): Busca un vehículo por matrícula en el concesionario.
- **Razón de Uso:** Ofrece métodos para gestionar la información de los vehículos en el concesionario.
- 3. **Clase Vehiculo:**
 - **Métodos utilizados:**
 - toString(): Proporciona una representación de cadena del objeto Vehículo.
 - **Razón de Uso:** Facilita la visualización de la información de un vehículo.
- 4. **Clase Propietario:**
 - **Propósito:** Representa al propietario de un vehículo.
 - **Razón de Uso:** Organiza la información del propietario para un mejor manejo.
- 5. **Clase UtilComprobaciones:**
 - **Métodos utilizados:**
 - Varios métodos de validación (para cantidades, enteros positivos, formato DNI, matrícula, nombre completo).
 - **Razón de Uso:** Proporciona funciones de validación que se utilizan en diferentes partes del programa, asegurando la integridad de los datos ingresados.
- 6. **Clase UtilDatos:**
 - **Métodos utilizados:**
 - entradaTeclado(String mensaje): Obtiene la entrada del usuario desde el teclado.
 - crearPropietario(): Crea un objeto Propietario a partir de la entrada del usuario.
 - crearVehiculo(Concesionario concesionario): Crea un objeto Vehiculo a partir de la entrada del usuario.
 - **Razón de Uso:** Facilita la interacción con el usuario y la creación de objetos Propietario y Vehiculo.
- 7. **Clase MostrarDatos:**
 - **Métodos utilizados:**
 - Varios métodos para mostrar información sobre vehículos, como listar vehículos, buscar por matrícula, modificar kilómetros y más.
 - **Razón de Uso:** Proporciona funciones para mostrar información específica sobre vehículos y realizar operaciones relacionadas con la presentación de datos.
- 8. **Clase Main:**
 - **Métodos utilizados:**
 - main(String[] args): Punto de entrada principal para la aplicación.
 - Varios métodos que interactúan con el usuario para agregar, listar, buscar, modificar y borrar vehículos.
 - **Razón de Uso:** Controla el flujo principal del programa y utiliza otras clases y métodos para realizar acciones según la elección del usuario.

En resumen, he diseñado y organizado el código de esta manera para mejorar la legibilidad, mantenimiento y extensibilidad del programa, manteniendo un enfoque claro en la gestión de vehículos en un concesionario. Cada clase y método tiene un propósito específico y contribuye a la funcionalidad general del programa.