Práctica de Programación Orientada a Objetos
Curso 2016-2017

Teodoro Ricardo García Sánchez
Dni: 50450469S

Introducción:

La aplicación consiste en desarrollar en java (entorno bluej) un programa de gestión de un taller de coches de acuerdo a las instrucciones entregadas por el equipo docente. Existen diferentes niveles de implementación. Esta práctica está completa hasta el nivel de implementación 3.a

Análisis y diseño:

He basado el programa en los siguientes conceptos:
Una clase taller que contiene toda la información necesaria para la gestión del taller. Así como unos métodos para poder manipular los datos.
Una clase ConsoleUI que contiene todos los menús necesarios para que los usuarios el sistema puedan realizar las acciones.

Clases:

Breve descripción de las clases:
Clase taller: Contiene toda la lógica necesaria para el manejo del taller.
Contiene los repositorios conteniendo los diferentes objetos necesarios para almacenar los datos
Se inyecta a la clase ConsoleUI para que pueda hacer uso de los métodos públicos.
En caso de existir una interfaz gráfica sería muy sencillo reutilizar toda la lógica, e incluso poder usar uno u otro dependiendo del caso.

Clase ConsoleUI:

Contiene toda la interfaz de usuario necesaria para manejar la aplicación.

Clase Vehículo:

Es la clase base de cualquier tipo de vehículo. Contiene un método llamado getTareasDeMantenimiento que se puede sobrescribir por las clases hijas para devolver las tareas de mantenimiento propias de cada vehículo. Por ejemplo, dependiendo del número de ruedas o de si el vehículo es de emergencias o no.

Clase Usuario:

Es la clase base de cualquier tipo de usuario. Los distintos tipos de usuario se almacenan en repositorios distintos. Por ejemplo, los clientes se almacenan en una Lista.

Clase Revisión:

Es la clase que almacena los datos necesarios para una revisión.
Almacena el vehículo, el cliente, el mecánico asignado y el estado en el que se encuentra la revisión.

Métodos públicos más relevantes:

Clase taller:

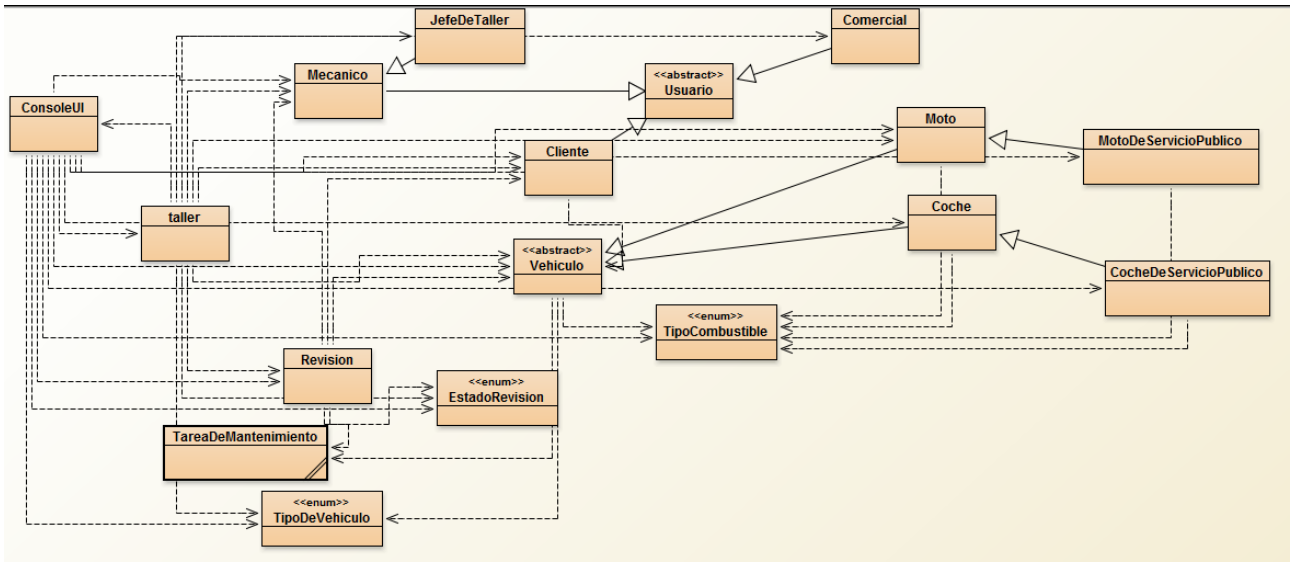
main: *Crea una instancia de la clase taller y se la inyecta a la clase ConsoleUI. A su vez invoca el método "mainMenu" que lanza el menú principal.*
darDeAltaVehiculo: *Da de alta un vehículo.*
darDeAltaCliente: *Da de alta un cliente.*
darDeAltaRevision: *Da de alta una revisión*
obtenerListaVehiculos: *Devuelve la lista de vehículos.*
obtenerListaClientes: *Devuelve la lista de clientes.*
obtenerSiguienteRevision: *Devuelve la siguiente revisión para el mecánico.*
obtenerListaRevisionesPendientes: *Devuelve la lista de revisiones pendientes.*
obtenerListaRevisionesRealizadas: *Devuelve la lista de revisiones realizadas.*
obtenerListaVehiculosRevisados: *Devuelve la lista de vehículos revisados.*
obtenerListaRevisiones: *Devuelve la lista de revisiones pendientes.*
getListaMecanicos: *Devuelve la lista de mecánicos dados de alta en el sistema.*

Clase Vehiculo:

getTareasDeMantenimiento: Devuelve las tareas de mantenimiento específicas del tipo de vehículo.

Aparte de estos métodos existen getters y setters donde corresponda.

DIAGRAMA DE CLASES



DOCUMENTACIÓN APORTADA

- Esta memoria
- Código fuente del programa: documento anexo_codigo.pdf
- Diagramas de clases: documentos tetris.png y pieza.pg
- Archivos .java y .class: codigo de la aplicación, en la carpeta src