

Laboratorio de programación 1

Trabajo Práctico N°3

1. Modelizar UML de las clases pedidas con sus relaciones.
2. Generar los archivos pedidos de las clases que considere necesarias e implementar un main con las características mencionadas al final del archivo.
3. Generar los métodos `to_string` e `Imprimir` en pantalla para cada clase, ninguna debe recibir nada por parámetro y devuelve los atributos concatenados como un string
4. Generar los getters y setters que crea necesarios!!



**El código deberá ser subido en github y no se aceptarán
archivos comprimidos**

**¡LOS MÉTODOS NO DEBEN RECIBIR NI ESCRIBIR EN
CONSOLA!** (Salvo método imprimir y listar y si se aclara
específicamente).

Se desea implementar un sistema para la administración de una compañía de alquiler de vehículos. La empresa cuenta con una lista de vehículos, una lista de clientes, y una lista de alquileres. Además de alquilar los vehículos, la empresa tiene la capacidad de adquirir nuevos vehículos, de retirarlos de circulación y de realizarles mantenimiento.

Todos los vehículos cuentan con número de patente, color, número de chasis, número de póliza, fecha de último mantenimiento, precio de alquiler por día, capacidad de pasajeros.

Hay 4 tipos de vehículos (**Clase abstracta!!**): motocicletas, automóviles, camionetas, y trafics. El precio de alquiler por día es el mismo para toda la categoría (es decir: todas las motos salen lo mismo) Para todos los vehículos es necesario calcular el costo final de alquiler que consta de una tarifa base + un precio por día (Método Calcular tarifa).

Además, para cada vehículo se pueden añadir elementos de seguridad u otros optativos:

- Las motocicletas pueden añadir el alquiler de uno o dos cascos.
- Los autos pueden alquilar sillas de seguridad para niños.
- Las camionetas pueden alquilar sillas de seguridad para niños y/o portaequipaje adicional.
- Las trafics pueden alquilar sillas de seguridad para niños y/o asientos rebatibles que pueden instalarse en el pasillo a forma de asiento auxiliar.

Los adicionales no agregan a la tarifa base, solamente al precio por día.

Para el mantenimiento, cada vehiculo tiene una verificación diferente (**Recuerden que NO debe imprimir en pantalla nada este método, solamente modificar algún valor DIFERENTE en cada tipo de vehiculo**) **Queda en la imaginación del grupo como integran el mantenimiento.**

Se necesita que el sistema tenga un método PasosMantenimiento que imprima los pasos claves para el mantenimiento de cada vehículo.(Recordar también cambiar la fecha del último mantenimiento).

En los alquileres se detalla el cliente, el vehículo, la cantidad de cada uno de los elementos de seguridad/adicionales involucrados, la fecha de inicio y de fin de la reserva y el monto total.

Se tiene una lista de alquileres que almacene todos los alquileres que existen.

Se debe poder listar todos los alquileres por tipo de vehiculo y calcular la ganancia total de eso. (**Ayuda: Utilizar una clase que hereda de la lista template y extenderla para que permita esto**).

Sobrecargar el operador de impresión para poder imprimir los datos de un vehiculo.

Sobrecargar el método suma para poder agregar items a una lista.

Sobrecargar el operador corchetes para acceder a un elemento de la lista de una cierta posición.

Utilizar **como mínimo un dynamic cast y un método virtual**.

IMPORTANTE, Esta consigna es la base de lo que necesitan hacer, cualquier cosa extra es bienvenida o si les interesa agregar algún concepto que hayan visto en algún lado pueden hacerlo, en caso que hagan eso no será considerado para la corrección, pero sí como

concepto. En caso de que haya problemas con lo que le hayan querido agregar (Por ejemplo una librería gráfica, de sonido, etc) tendrán prioridad los que tengan problemas con la consigna original.

Genere el main y utilice todos sus métodos para verificar su funcionamiento creando todas como objetos dinámicos (Recuerde liberar la memoria al creer necesario).