

NOMBRE

Para hacer el examen debes conectarte con el usuario que se te indique.

Utilizad las clases Fecha y Leer para las entradas de datos. No se podrán utilizar otros recursos de la API-8 de java distintos a los que se han explicado hasta la presente evaluación.

Se atenderá a los principios de ocultamiento y encapsulación de la POO; es decir, sin variables globales y los métodos solo pueden manejar variables locales y los parámetros que reciban.

Al terminar el examen envía por correo electrónico un archivo comprimido con el contenido de los dos proyectos indicados a continuación. El archivo comprimido se llamará **TuApellido2ev.zip** (por ejemplo Perez2ev.zip). Las direcciones donde debes enviar el correo son:

m.a.alvarez@iessantiagohernandez.com in1nle@gmail.com

Es responsabilidad tuya enviar correctamente la solución.

1) Crea un proyecto denominado ProyectoMuebles. (5 puntos)

Escoge uno de los dos modelos de clases a implementar propuestos para resolver el siguiente problema:

a) Diseña una clase `Mueble` que cumpla las siguientes especificaciones:

Tendrá los siguientes atributos que **no** deben ser accesibles fuera de la clase:

- `codigo` (Integer autoincremental)
- `descripcion` (de la clase String)
- `ancho`
- `alto`
- `fondo`

Y los siguientes métodos:

- Un método constructor que reciba los parámetros descripción, ancho, alto y fondo.
- Un método, que se llamará `muestraMueble()` que facilite la información del mueble: (utiliza como ejemplo el método `toString()`).

b) Escribe una clase `Almacen` que cumpla las siguientes especificaciones:

Tendrá los siguientes atributos que **no** deben ser accesibles fuera de la clase:

- `nombre`
- `direccion`

Y los siguientes métodos:

- Un método constructor que reciba los parámetros nombre y direccion.
- Un método, que se llamará `muestraAlmacen()` que facilite la información del almacén: (utiliza como ejemplo el método `toString()`).

c) Los muebles se guardan en los diferentes almacenes. Según el diagrama de clases elegido crea el método correspondiente para esta asignación.

d) Crea una clase que se llame `GestionMueble` en la que esté el método `main` y los siguiente métodos:

- un método que instancie 4 objetos de clase `Almacen` y 15 objetos de tipo `Mueble`.
- un método que muestre en pantalla los 15 muebles.
- un método que muestre en pantalla los 4 almacenes
- un método que muestre en pantalla la información de cada almacén con la relación de muebles que se guardan en él.

La nota de la 2ª evaluación se calculará: $\text{nota} = 0.3 \text{ examen_test} + 0.7 \text{ examen_problemas}$

Siempre y cuando se obtenga al menos un 4 en cada examen

NOMBRE

2) Crea un proyecto denominado **ProyectoCoches**. (5 puntos)

Utiliza el código facilitado de la clase `GestionAgencia` y el diagrama de clases que se te ha entregado.

Vamos a diseñar un programa en Java para llevar el control de una agencia de coches de alquiler. Todos los atributos que se usen en las clases deben ser privados. Los métodos `set` deben recibir el parámetro correspondiente (dentro de un método `set` no se puede pedir nada por teclado) y los métodos `get` devolver el atributo al que se refieran (no se puede mostrar nada en pantalla).

Hay **coches** de dos categorías: **normal** y de **lujo**.

De todos los coches nos interesa conocer la matrícula, marca, modelo, fecha de fabricación, si está disponible u ocupado y precio por día.

Los coches de lujo tienen un código numérico que empezará en 10001 para el primer coche de lujo que se dé de alta en la empresa y se incrementará en 1 para cada nueva alta. Este código, que se pega en el parabrisas del vehículo, da derecho a llenar gratis una vez el depósito de combustible durante los días que se tiene alquilado el coche. Lo utilizan en las estaciones de servicio con las que tiene convenio la agencia para llevar el control de los coches que aprovechan la oferta.

El coche normal tiene además, un `String` cuyo valor puede ser: "Aparcamiento del Ebro", "IberPark" o "Aragonia", que son los nombres de las empresas que gestionan aparcamientos públicos que permiten aparcar a los coches de la agencia aplicando un descuento del 50%.

Cuando un cliente **alquila** un vehículo indica si lo quiere normal o de lujo y, si hay alguno disponible, se marca el coche como ocupado y ya se lo puede llevar.

Cuando lo **devuelve** se solicita el número de días que lo ha tenido alquilado, se marca el coche como **disponible** y se muestra en pantalla el importe que debe pagar.

Diseña las clases que hay utilizar para desarrollar la aplicación, teniendo en cuenta que en la agencia hay 15 coches normales y 5 de lujo.

La clase que tiene el método `main` se llamará `GestionAgencia`. Esta clase tendrá un método estático, al que se llamará justo después de declarar las variables del método `main`, que se encargará de crear los datos de los 20 coches asignando valores aleatorios a las matrículas (4 números y 3 letras mayúsculas) y a los precios. El precio diario de un coche normal oscilará entre 20 y 30 € y el de un coche de lujo entre 50 y 60 €.

La clase `GestionAgencia` mostrará el siguiente menú:

- 1- Alquilar coche.** Hay que filtrar que el coche que se quiere alquilar está disponible.
- 2- Devolver coche.** Hay que filtrar que el coche que se quiere devolver estaba alquilado.
- 3- Listado de coches disponibles.**
- 4- Listado de coches ocupados.**
- 0- Salir.**

Los listados sacarán para cada tipo de coche toda su información, una línea para cada vehículo.

El programa se refactorizará lo más que se pueda.