

**EXAMEN UNIDAD 4 TIPO D**
**NOMBRE:** \_\_\_\_\_ **FECHA:** \_\_\_\_\_

Resultados de aprendizaje	Criterios de evaluación	Calificación (cada criterio se puntúa sobre 10)
4. Desarrolla programas organizados en clases utilizando las características principales del lenguaje de programación (Interfaces, herencia, polimorfismo...)	4.2. Se han creado y utilizado métodos estáticos y/o reescritos	
	4.4. Se ha utilizado el polimorfismo de forma adecuada	
	4.5. Se han distinguido instancias de diferentes clases y usado el casteo de objetos e interfaces cuando ha sido necesario	

1. En las películas de ciencia ficción existen muchos tipos de vehículos y siempre me he preguntado cómo sería tener uno. Me gustaría montar una empresa que se dedique a alquilar este tipo de vehículos. Os cuento que debería hacer el programa: Deberían poderse alquilar Vehículos Genéricos y otros de distinto tipo Batmóviles y Patinetes Voladores.

Para cada ALQUILER se guarda el nombre de la persona que lo hace, su DNI y el número de días de alquiler. Un VEHÍCULO genérico se caracteriza por su matrícula, su tamaño en metros y año de fabricación.

El precio de un alquiler de un vehículo genérico se calcula multiplicando el su tamaño en metros por el año de fabricación. Esta cantidad serían céntimos por lo que hay que dividirla por cien para dar el precio en euros.

Sin embargo ahora se pretende diferenciar la información de algunos tipos de vehículos “especiales”:

- Número de ruedas para Batmóviles.
- Potencia en caballos para patinetes voladores de regreso al futuro.

El alquiler de los vehículos de un tipo *especial* se obtiene como el de uno normal **más**:

- ✓ El número de ruedas para Batmóviles por dos euros.
- ✓ Un euro por cada caballo para patinetes del regreso al futuro.

Por otro lado, la clase Vehículo debe implementar la siguiente interfaz y el método debe ser reescrito en las clases que sea necesario.

```
public interface IAlquiler {

    public double calcularPrecio ( );

}
```

Cree una clase **Alquiler** con un único atributo que será un array de tipo Vehiculo y tendrá los siguientes métodos:

\* calcularAlquilerUnVehiculo (Vehiculo v); Buscar el vehículo de la manera que tú consideres.

\* mostrarListado ( ) mostrando todos los datos de todos los objetos y que cuando en el listado aparezca un patinete se muestre el mensaje de “¡Cuidado que voy volando!”. Para ello, se debe tener un método solo en la clase Patinete llamado avisarPatinete ( ), que muestre dicho mensaje de aviso solo en ese caso.

\* calcularTotalRecaudado ( );

\* calcular el total recaudado solo con los Batmóviles.

Para probar todo esto, se debe crear en el main un array en cuyo interior se guarde, al menos, un vehículo de cada tipo (incluidos los vehiculos en general) y realizar mostrar las siguientes cosas:

Se pueden crear los objetos directamente con los valores como parámetros.

Crear en el main los objetos necesarios, rellenar el array con objetos de todos los tipos posibles (se pueden crear los objetos directamente con los valores como parámetros) y hacer las llamadas correspondientes para poder ver toda la funcionalidad del programa.

**NOTA: No es necesario un menú, solo se debe hacer este si hay tiempo al final.**

**Si el programa no compila, por cualquier motivo, se dividirá la nota por dos. Criterios de corrección. Cada criterio se puntúa sobre 10.**

**4.2. Reescritura adecuada de métodos, 7 puntos. Llamada a dicho método, 3 puntos.**

**4.4. Polimorfismo en profundidad, 10. Polimorfismo a menor nivel, 6 puntos.**

**4.5. Uso de las distintas clases, interfaz, instanceof y casteo, 10 puntos.**