

CICLO FORMATIVO: DAM  
MÓDULO PROFESIONAL: Programación  
FECHA Y HORA: Martes 7 de junio 18:45  
DURACIÓN: 2 horas 45 minutos

APELLIDOS, NOMBRE:		DNI:	
IES de referencia:	<input type="checkbox"/> IES Aguadulce (Almería) <input type="checkbox"/> IES Trassierra (Córdoba)		
IES donde se realiza el examen:			

**INSTRUCCIONES:**

**Selecciona tu centro de referencia y profesorado asignado:**

IES Aguadulce	DAM	<input type="checkbox"/> Mila	<input type="checkbox"/> Fran	<input type="checkbox"/> Germán
IES Trassierra	DAM	<input type="checkbox"/> David	<input type="checkbox"/> Manuel	

**El examen práctico se puede realizar con todo el material que el alumnado estime oportuno (apuntes, libros, conexión a Internet y a la plataforma, pendrive, disco duro externo,...)**

**Se prohíbe el uso del móvil o de su propio portátil**, debiendo hacerse con el ordenador que le facilita el centro, que debe disponer de conexión a Internet.

- El examen consta de 4 ejercicios, el 1 y el 2 corresponden al primer cuatrimestre y el 3 y 4 al segundo.
- **Puntuación: 10 puntos en total (Todos los ejercicios tienen la misma puntuación).**
- Si se ha superado el examen de febrero (nota mayor o igual a 5), la nota mínima del examen para hacer media con el de febrero debe ser de 4 puntos.
- **El alumnado que superó el examen de febrero solo tiene que hacer los ejercicios 3 y 4 (excepto si quiere subir nota que deberá hacerlos todos).** El resto del alumnado debe hacer los cuatro ejercicios.
- **El examen se desarrollará a partir del proyecto base facilitado en la plataforma, para facilitar su realización y corrección. Una vez realizado se subirá en el buzón habilitado al efecto, como cualquier otra tarea.**

**CALIFICACIÓN:**

Mediante esta prueba se evalúan los resultados de aprendizaje y criterios de evaluación del módulo que se detallan a continuación:

RESULTADOS DE APRENDIZAJE	CRITERIOS DE EVALUACIÓN	PONDERACIÓN
RA1	a)-i)	9 %
RA2	a)-i)	11 %
RA3	a)-g)	16 %
RA4	a)-j)	17 %
RA5	a)-h)	12 %
RA6	a)-i)	15 %
RA7	a)-h)	15 %

<b>Peso total de la prueba en la calificación del módulo:</b>	<b>40 %</b>
---	-------------

## EXAMEN:

### 1.- Dado el array proporcionado con diversas cadenas correspondientes a matrículas de vehículos

```
String[] arrayMatriculas = {"13", "MU9142BN", "1A", "0", "1",  
"AB2747T", "AL7347T", "FF", "63", "111", "AB1234YU", "M5723P", "7F"};
```

Una matrícula se considerará válida si cumple las siguientes condiciones:

- comienza por el prefijo "AB" o "MU";
- a continuación le siguen cuatro dígitos numéricos;
- termina con una o dos letras mayúsculas (cualquier letra, sin restricción).

Escribe un programa que se encargue de:

- Indicar qué matrículas son válidas según el criterio anterior empleando una expresión regular.
- Calcular el número total de válidas y no válidas.

#### MUESTRA DE SALIDA DEL EJERCICIO 1

Matrículas válidas
-----
MU9142BN
AB2747T
AB1234YU
Total de matrículas válidas: 3
Total de matrículas no válidas: 10

### 2.- Dado un array de cadenas que debería contener números reales

```
String[] arrayCadenas = {"xxx", "2019", "23,5", "323.78", "2ab33",  
"8921.8", "234.8556", "Ea56"} ;
```

hacer un programa que lo recorra y analice cada elemento. En aquellos casos en los que se trate de un número válido debe mostrarse por consola con una precisión de dos decimales y una anchura de columna de tamaño 8.

#### MUESTRA DE SALIDA DEL EJERCICIO 2

CONTENIDO INICIAL DEL ARRAY
-----
[xxx, 2019, 23,5, 323.78, 2ab33, 8921.8, 234.8556, Ea56]
Número correcto: 2019,00
Número correcto: 323,78
Número correcto: 8921,80
Número correcto: 234,86

### 3.- Implementar la clase abstracta *Vehiculo* y dos clases hijas: *Coche* y *Bicicleta*

La clase *Vehiculo* dispondrá de los siguientes atributos:

- `codigo`, de tipo entero;
- `nombre`, de tipo `String`;
- `numRuedas`, de tipo entero.

Así mismo, también contará con:

- un **constructor** básico de tres parámetros: `Vehiculo(int codigo, String nombre, int ruedas)`;
- un método `toString` que devolverá una cadena con una representación textual del estado del objeto con el formato: "Código: **xxx** nombre: **yyy** con: **zzz** ruedas";
- un **método abstracto** `getVelocidadMaxima()`, sin parámetros, que devolverá un `float`.

La clase *Coche* incluirá un atributo entero para indicar el **número de cilindros**, junto con:

- un **constructor** de tres parámetros: `Coche(int codigo, String nombre, int cilindros)`;
- un método `toString` que devolverá una cadena con una representación textual del estado del objeto con el formato: "Código: **xxx** nombre: **yyy** con: **zzz** ruedas con **vvv** cilindros";
- la **implementación de los métodos abstractos** de aquellos métodos que estén sin implementar en sus ancestros.

Debes tener en cuenta que todo objeto instancia de la clase *Coche* siempre tendrá **cuatro ruedas** y que la **velocidad máxima** de un coche se calculará multiplicando su cantidad de cilindros por 100 y dividiendo por 1,85

La clase *Bicicleta* incluirá un atributo entero para indicar el **número de radios**, junto con:

- un **constructor** de cuatro parámetros: `Bicicleta(int codigo, String nombre, int ruedas, int numRadios)`;
- un método `toString` que devolverá una cadena con una representación textual del estado del objeto con el formato: "Código: **xxx** nombre: **yyy** con: **zzz** ruedas número de radios: **vvv**";
- la **implementación de los métodos abstractos** de aquellos métodos que estén sin implementar en sus ancestros.

Debes tener en cuenta que todo objeto instancia de la clase *Bicicleta* siempre tendrá **dos ruedas** y que la **velocidad máxima** de una bicicleta se calculará multiplicando su cantidad de radios por 1,75.

Por último, habrá que completar el `main` del programa principal para crear un coche y una bicicleta con los datos que se observan en la siguiente salida de ejecución.

MUESTRA DE SALIDA DEL EJERCICIO 3:

```
Crear un coche
El primer coche es: Código: 1 nombre: Seat Fura con: 4 ruedas con 4
cilindros.
Y tiene una velocidad máxima de : 216.21622 kms. por hora

Crear una bicicleta
La primera bicicleta es: Código: 4 nombre: Bicicross BH con: 2 ruedas
número de radios: 28
Y tiene una velocidad máxima de : 49.0 kms. por hora
```

4.- Dados el archivo de texto “*coches.txt*” y la clase **Coche** que ya se proporcionan en el ejercicio:

- Crear un **Map** para almacenar los valores leídos. Las claves serán de tipo entero y consistirán en un números consecutivos a partir del 1. Los valores serán referencias a objetos instancias de la clase **Coche**.
- Abrir el archivo *coches.txt* y, para cada línea leída, crear un objeto coche con el valor leído y almacenar en el **Map** el par <clave, valor> correspondiente (contador y objeto **Coche** recién creado).
- Una vez rellenado el **Map**, recorrer todos sus pares <clave, valor> y mostrarlos por pantalla.

MUESTRA DE SALIDA DEL EJERCICIO 4:

```
-- Lectura de fichero de Texto --
-----

----- Contenido del mapa -----
Clave: 1 -> Valor: Coche: Citroen Xsara Picasso
Clave: 2 -> Valor: Coche: Seat 600
Clave: 3 -> Valor: Coche: Renault Megane CC
Clave: 4 -> Valor: Coche: Mercedes clase A
Clave: 5 -> Valor: Coche: Fiat 500
Clave: 6 -> Valor: Coche: Renalt Copa Turbo
Clave: 7 -> Valor: Coche: KIA karateka
```