

PROGRAMACIÓN – 1º D.A.W. IES DOMENICO SCARLATTI EXAMEN 2ª EVALUACIÓN – PARTE 2		Tipo: B
APELLIDOS:	Calificación:	
NOMBRE:		
FECHA: 05-02-2021		

PUNTUACIÓN: 5 puntos. TIEMPO: 60 minutos

Importante: indenta (sangra) el código para mejorar su legibilidad. No sangrar el código supone una penalización de -0'25 en cada ejercicio.

- Codifica el método:

```
public static int[] copiaArray (int[] num, int ini, int fin)
```

que recibe un array de enteros y devuelve un array que es una copia que contiene los elementos desde el índice ini hasta el índice fin (incluido). **(1 punto)**
- Codifica el método:

```
public static void imprimeSumaColumnas (double[][] m, int filas, int col)
```

que recibe una matriz bidimensional e imprime la suma de cada columna de la matriz (comenzando por la columna cero).

Por ejemplo, si la matriz que le pasásemos al método fuese:

```
2.0    3.0    4.0
1       7       7
10     20.0   30
```

el método imprimiría:

```
13.0
30.0
41.0
```

(1 punto)
- La clase `Empleado` debe guardar la siguiente información de cada empleado de la empresa: DNI (sin letra), nombre, apellidos, salario bruto anual y si tiene reducción de jornada o no.

Cuando se creen objetos de la clase `Empleado`, se le proporcionará obligatoriamente el DNI. No se puede crear esta clase si no le pasamos el DNI al crear el objeto.

El DNI y el salario no pueden ser accedidos directamente (ni para leer ni para modificar) por otras clases. Los demás atributos, sí. Tendrás que proporcionar para las clases externas un método que te devuelva el DNI y otros dos que te permitan obtener el salario y modificarlo (asignar un nuevo valor al salario del empleado).

También sobrescribiremos el método `toString()` que devolverá el valor del DNI y el nombre y apellidos del empleado. Se pide:
 - Codifica la clase `Empleado` con los atributos y métodos necesarios. Debes elegir los tipos de datos adecuados para cada atributo y para los parámetros y valores devueltos por cada método, así como los modificadores de acceso (`public` o `private`) de los atributos y métodos.
 - Codifica la clase `PruebaEmpleado` en la que crearás un objeto de la clase `Empleado`, darás valores a todos los atributos e imprimirás finalmente información del empleado utilizando el método `toString()`.
(1'25 puntos)

4. La clase abstracta `Vehículo` tiene como atributo el número de kilómetros recorridos por el vehículo y los métodos `repostar()`, que simplemente imprime “Vehículo repostando” y `pitar()` que imprime un mensaje.
Las clases `Coche` y `Furgoneta` heredan de `Vehículo` y sobrescriben el método `pitar()`.
Tanto `Coche` como `Furgoneta` codifican los métodos del interfaz `Conducible`:
`arrancar()`, `acelerar()` y `parar()`.

Nota: los métodos lo único que hacen es mostrar un mensaje identificativo por pantalla.

Se pide:

- a. Codifica todas las clases y el interfaz especificado (no hace falta hacer métodos set/get ni constructores)
- b. Codifica la clase `Prueba` en la que crees dos objetos `Coche` y dos objetos `Furgoneta` y los almacenes (a los cuatro) en un array para conducirlos. Deberás recorrer el array para arrancar todos los vehículos que hayas almacenado.
(1'25 puntos)

5. Realiza un método:

```
public static int leeNumeros();
```

que pida números enteros al usuario hasta que éste introduzca cero y devuelva la suma de todos los números leídos. La solución debe ser **recursiva**, es decir, se leerá un solo número en cada llamada a la función: no se pueden utilizar estructuras iterativas como `while` o `for`. No se valorarán soluciones no recursivas.

Supón que la función puede utilizar, sin tener que crearlo, el siguiente objeto de la clase `Scanner`, para pedir los datos al usuario:

```
private static Scanner lector;
```

(0'5 puntos)