

Familia Profesional Informática y Telecomunicaciones		Nombre del Ciclo Formativo Título de Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Centro Educativo IES Campanillas (sede Marie Curie)		Módulo Profesional Programación Código: 0485 N.º de créditos ECTS: 14	Profesor Luis José Sánchez González	
Curso lectivo 2017 / 2018	Grupo 1º DAW	Tipo de documento Examen	Trimestre Segundo – Control 1	Fecha 20 de febrero de 2018

INSTRUCCIONES

- ➔ El alumno debe entregar una carpeta con las soluciones al examen cuyo nombre debe estar formado por "Ex" seguido del número de lista, seguido de las iniciales. Por ejemplo, Facundo Romuedo Piladro que es el número 8 de la lista entregaría una carpeta con nombre **Ex08frp**.
- ➔ Los ficheros o carpetas correspondientes a las soluciones se deben nombrar igual que la carpeta junto con el número del ejercicio, por ejemplo **Ex08frp1.java, Ex08frp2.java, etc.**
- ➔ En los comentarios de cada programa **se debe indicar el nombre completo**, la fecha y - si procede - el turno.
- ➔ Únicamente se necesita entregar el código fuente en java, **no se deben entregar los archivos con la extensión .class**.

EJERCICIOS

1. Implementa la clase **Terminal**. Un terminal tiene asociado un número. Los terminales se pueden llamar unos a otros y el tiempo de conversación corre para ambos. A continuación se proporciona el contenido del main y el resultado que debe aparecer por pantalla.

```
Terminal t1 = new Terminal("678 11 22 33");
Terminal t2 = new Terminal("644 74 44 69");
Terminal t3 = new Terminal("622 32 89 09");
Terminal t4 = new Terminal("664 73 98 18");
System.out.println(t1);
System.out.println(t2);
t1.llama(t2, 320);
t1.llama(t3, 200);
System.out.println(t1);
System.out.println(t2);
System.out.println(t3);
System.out.println(t4);
```

```
Nº 678 11 22 33 - 0s de conversación
Nº 644 74 44 69 - 0s de conversación
Nº 678 11 22 33 - 520s de conversación
Nº 644 74 44 69 - 320s de conversación
Nº 622 32 89 09 - 200s de conversación
Nº 664 73 98 18 - 0s de conversación
```

2. Implementa la clase **Movil** como subclase de **Terminal** (esta clase no se modifica). Cada móvil lleva asociada una tarifa que puede ser **"rata"**, **"mono"** o **"bisonte"**. El coste por minuto es de 6, 12 y 30 céntimos respectivamente. Se tarifican los segundos exactos. Obviamente, cuando un móvil llama a otro, se le cobra al que llama, no al que recibe la llamada. A continuación se proporciona el contenido del main y el resultado que debe aparecer por pantalla.

```
Movil m1 = new Movil("678 11 22 33", "rata");
Movil m2 = new Movil("644 74 44 69", "mono");
Movil m3 = new Movil("622 32 89 09", "bisonte");
System.out.println(m1);
System.out.println(m2);
m1.llama(m2, 320);
m1.llama(m3, 200);
m2.llama(m3, 550);
System.out.println(m1);
System.out.println(m2);
System.out.println(m3);
```

```
Nº 678 11 22 33 - 0s de conversación - tarificados 0.00 euros
Nº 644 74 44 69 - 0s de conversación - tarificados 0.00 euros
Nº 678 11 22 33 - 520s de conversación - tarificados 0.52 euros
Nº 644 74 44 69 - 870s de conversación - tarificados 1.10 euros
Nº 622 32 89 09 - 750s de conversación - tarificados 0.00 euros
```

3. Realiza una función que tome como parámetros dos arrays de cadenas de caracteres y que devuelva otro array con los valores comunes a ambos. Se distinguen mayúsculas de minúsculas, por tanto "hola" es distinto de "Hola". Por ejemplo, si el array **a** contiene los valores {"casa", "coche", "sol", "mesa", "ordenador", "sol", "CASA"} y el array **b** contiene los valores {"zambomba", "coche", "casa", "sol", "mermelada"}, la ejecución de **palabrasComunes(a, b)** devolvería el array {"casa", "coche", "sol"}. Para simplificar el ejercicio, se puede suponer que los arrays que se pasan como parámetros no contienen valores repetidos. Se debe entregar tanto el código de la función como el código de prueba que la usa. La cabecera de la función es la siguiente:

```
public static String[] palabrasComunes(String[] a, String[] b)
```

4. Implementa una función con nombre **nucleo** que sea capaz de extraer el array de 2 filas por 2 columnas que hay dentro de un array cuadrada. Si el array que se pasa como parámetro no es cuadrado o no tiene como mínimo 4 elementos o tiene un número de filas/columnas impar, la función debe devolver un array de 2 filas por 2 columnas rellena con -1, esto es {{-1, -1}, {-1, -1}}. Se debe entregar tanto el código de la función como el código de prueba que la usa. La cabecera de la función es la siguiente:

```
public static int[][] nucleo(int[][] n)
```

Ejemplo 1:

Si el array **a** es el que se muestra a continuación:

```
35 72 24 45 42 60
32 42 64 23 41 39
98 45 94 11 18 48
12 34 56 78 90 12
```

El array devuelto por **nucleo(a)** sería el siguiente:

```
-1 -1
-1 -1
```

Ejemplo 2:

Si el array **b** es el que se muestra a continuación:

```
45 92 14 20 25 78
23 12 94 85 76 20
45 92 14 20 25 78
35 72 24 45 42 60
32 42 64 23 41 39
98 45 94 11 18 48
```

El array devuelto por **nucleo(b)** sería el siguiente:

```
14 20
24 45
```