

Familia Profesional Informática y Telecomunicaciones		Nombre del Ciclo Formativo Título de Técnico Superior en Desarrollo de Aplicaciones Multiplataforma / Web		
Centro Educativo IES Campanillas - PTA		Módulo Profesional Programación Código: 0485 N.º de créditos ECTS: 14	Profesor Luis José Sánchez González	
Curso lectivo 2017 / 2018	Grupo 1º DAM y 1º DAW	Tipo de documento Examen	Trimestre Segundo – Recuperación	Fecha 22 de marzo 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**.

1. Realiza una función con nombre **linea** que sea capaz de pintar una línea horizontal de caracteres por pantalla. La longitud de la línea y el carácter con el que se pinta se pasan como parámetros. Utiliza esta función para realizar el programa de la pirámide rellena, es decir, que pida la altura y muestre por pantalla una pirámide. La cabecera de la función es la siguiente:

```
public static void linea(int longitud, String caracter)
```

```
linea(0, "*") pinta por pantalla la cadena vacía
```

```
linea(1, "*") pinta por pantalla *
```

```
linea(5, "A") pinta por pantalla AAAAA
```

```
linea(7, "€") pinta por pantalla €€€€€€€
```

Ejemplo 1:

Introduzca la altura de la pirámide: 6

```

      *
    * * *
  * * * * *
* * * * * *
* * * * * * *
* * * * * * *
* * * * * * *

```

Ejemplo 2:

Introduzca la altura de la pirámide: 4

```

      *
    * * *
  * * * *
* * * * *

```

2. Implementa una función con nombre **nEsimo** que busque el número que hay dentro de un array bidimensional en la posición n-ésima contando de izquierda a derecha y de arriba abajo, como si se estuviera leyendo. El primer elemento es el 0. Si la posición donde se busca no existe en el array, la función debe devolver -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 nEsimo(int[][] n, int posicion)
```

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
```

```
nEsimo(a, 0) devuelve 35
nEsimo(a, 2) devuelve 24
nEsimo(a, 5) devuelve 60
nEsimo(a, 6) devuelve 32
nEsimo(a, 21) devuelve 78
nEsimo(a, 23) devuelve 12
nEsimo(a, 24) devuelve -1
nEsimo(a, 100) devuelve -1
```

3. La empresa **El Corte Inglés** nos ha encargado una aplicación para gestionar las tarjetas regalo. Como primer paso para implementar la aplicación, es necesario crear la clase principal. Implementa la clase **TarjetaRegalo**. Cuando se crea una nueva tarjeta, se le da un saldo y se asigna de forma automática un número de 5 dígitos. Si se intenta gastar más dinero del que tiene la tarjeta, se debe mostrar un mensaje de error. Dos tarjetas regalo se pueden fusionar creando una nueva tarjeta con la suma del saldo que tenga cada una y un nuevo número aleatorio de 5 cifras.

A continuación se proporciona el contenido del main y el resultado que debe aparecer por pantalla.

```
TarjetaRegalo t1 = new TarjetaRegalo(100);
TarjetaRegalo t2 = new TarjetaRegalo(120);
System.out.println(t1);
System.out.println(t2);
t1.gasta(45.90);
t2.gasta(5);
t2.gasta(200);
t1.gasta(3.55);
TarjetaRegalo t3 = t1.fusionaCon(t2);
System.out.println(t1);
System.out.println(t2);
System.out.println(t3);
```

```
Tarjeta nº 67324 - Saldo 100€
Tarjeta nº 02788 - Saldo 120€
No tiene suficiente saldo para gastar 200€
Tarjeta nº 67324 - Saldo 50.55€
Tarjeta nº 02788 - Saldo 115€
Tarjeta nº 59032 - Saldo 165.55€
```

4. Crea la clase abstracta **Figura** con los atributos **caracter** y **estaRellena** que indican el carácter con el que se debe pintar la figura y si está o no está rellena. Crea las clases **Cuadrado** y **Rectangulo** como subclases de **Figura**. La clase **Cuadrado** debe tener el atributo **lado** y la clase **Rectangulo**, los atributos altura y anchura. Implementa el método **toString()** para **Cuadrado** y para **Rectangulo** de tal forma que se puedan visualizar las figuras. Prueba las clases y los métodos desde un programa.

NOTAS: El examen se valorará **APTO** o **NO APTO**. El alumno que tenga que recuperar las dos partes, deberá tener bien, al menos, un ejercicio de cada parte para aprobar.