

Familia Profesional Informática y Telecomunicaciones		Nombre del Ciclo Formativo Título de Técnico Superior en Desarrollo de Aplicaciones Web		
Centro Educativo IES Campanillas (sede PTA)		Módulo Profesional Programación Código: 0485 N.º de créditos ECTS: 14	Profesor Juan Antonio Jiménez Morales	
Curso lectivo: 2019/2020	Grupo: 1º DAW	Trimestre: Terº – Control 8	Modelo: Único	Fecha: 21/05/2020 Pág. 1/8

EJERCICIOS DE POO SEÑALADOS EN AMARILLO

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. También debe indicar una breve descripción de lo que hace el programa.
- ➔ Únicamente se necesita entregar el código fuente en java, **no se deben entregar los archivos con la extensión .class**.

EJERCICIOS – Materia segundo trimestre

1. [4 puntos] Implemente las siguientes clases, con las siguientes especificaciones, referentes al juego del dominó:

- a) Clase **Ficha**: contendrá como atributos dos valores numéricos, cada uno de los cuales oscilará entre 0 y 6. La ficha la podemos considerar siempre en posición horizontal, teniendo un valor a la izquierda y otro a la derecha. Y tiene una orientación que viene dada por los valores que contiene.
- Al constructor se le pasarán los valores correspondientes a dicha ficha. Si se le pasan otros valores que no oscilen entre 0 y 6, aproximará al valor más cercano de los valores válidos.
 - El método **valorIzquierda** devuelve el valor situado a la izquierda de la ficha.
 - El método **valorDerecha** devuelve el valor situado a la derecha de la ficha.
 - El método **cambiarOrientacion** intercambiará los valores de dicha ficha (es como si le diera un giro de 180 grados),
 - El método **toString** devolverá la representación en cadena de la ficha, que consistirá en colocar entre paréntesis cuadrados y separados por una barra vertical los valores numéricos correspondientes a la ficha, con la excepción del valor 0, que se representará por un espacio en blanco. Ejemplo:

[3|4]

- El método **toHTML** devolverá la representación “en HTML” de la ficha, que consistirá en una secuencia de código HTML donde la ficha se representa por una tabla con una fila y dos columnas, con borde, y donde cada valor de la ficha se coloca en cada una de las celdas de la tabla. Las imágenes correspondientes a los valores de las fichas han de estar ubicadas en la carpeta “fichas”, que cuelga del directorio raíz del proyecto.

- b) Interfaz **ConjuntoDeFichas**, que exige que se implementen los siguientes métodos: **añadeFicha**, **toString** y **toHTML**, cuyo comportamiento se describe a continuación en las clases que implementan esta interfaz.
- c) Clase **Monton**: debe implementar la interfaz **ConjuntoDeFichas***, y contendrá como atributo un arraylist de fichas de la clase anterior. Un montón de fichas estará inicialmente vacío.
- El método **añadeFicha*** añadirá a este montón la ficha que se le pase como argumento. La añadirá siempre por el mismo extremo del montón. Siempre devolverá true. Este método viene obligado por la implementación de la interfaz.
 - El método **toString*** representará, como secuencia de caracteres, el estado del conjunto de fichas del montón, mostrando en varias líneas las fichas que contenga, a razón de 10 fichas por línea, y presentando las fichas con una separación de 2 espacios entre ellas. Ejemplo:

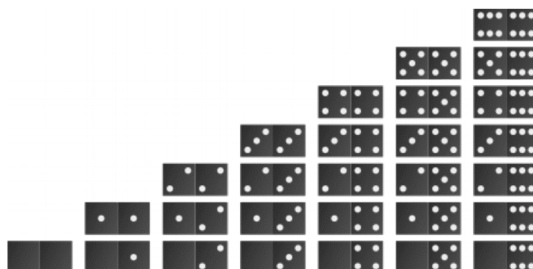
[3|4] [|6] [|]

- El método **toHTML*** representará, como secuencia de código HTML, el estado del conjunto de fichas del montón, mostrando las fichas que contenga, a razón de 10 fichas por línea, y presentando las fichas con una separación de 2 espacios entre ellas. Deberá utilizar tablas en HTML para lograr el efecto de disponer varias fichas en una fila.

- El método **sacaFichaAlAzar** devolverá una ficha aleatoriamente escogida, sacándola del montón. Si no hubiera fichas, devolverá null.

Familia Profesional Informática y Telecomunicaciones		Nombre del Ciclo Formativo Título de Técnico Superior en Desarrollo de Aplicaciones Web			
Centro Educativo IES Campanillas (sede PTA)		Módulo Profesional Programación Código: 0485 N.º de créditos ECTS: 14		Profesor Juan Antonio Jiménez Morales	
Curso lectivo: 2019/2020	Grupo: 1º DAW	Trimestre: Terº – Control 8	Modelo: Único	Fecha: 21/05/2020	Pág. 2/8

- El método **juntaMontones** saca las fichas del montón que se le pasa como argumento y las mete en el montón que recibe el método.
- El método **inicializaMonton** generará todas las fichas del dominó. Tenga en cuenta que las fichas del dominó son las siguientes (debe generarlas de manera automatizada; no de 1 en 1):



- d) Clase **Tablero**: debe implementar la interfaz **ConjuntoDeFichas***, y contendrá como atributo un arraylist de fichas de la clase anterior. El tablero estará inicialmente vacío.
- El método **añadeFicha*** añadirá al tablero la ficha que se le pase como argumento. La añadirá por el extremo que pueda, y si puede añadirla por los dos extremos, decidirá aleatoriamente por cuál de ellos la añade. Si la puede añadir, devolverá true; en otro caso, devolverá false. Este método viene obligado por la implementación de la interfaz.
- Obsérvese que si el tablero está vacío se puede añadir cualquier ficha.
- Para que una ficha pueda ser añadida por un extremo debe ocurrir que alguno de los valores de la ficha a añadir y el valor “que queda expuesto” en dicho extremo coincidan. Si fuera necesario, deberá “girar” la ficha.
- El método **toString*** representará, como secuencia de caracteres, el estado del conjunto de fichas del tablero, mostrando en varias líneas las fichas que contenga, a razón de 10 fichas por línea, y presentando las fichas sin separación entre ellas. Ejemplo:

[3|4][|6][|]

- El método **toHTML*** representará, como secuencia de código HTML, el estado del conjunto de fichas del tablero, mostrando las fichas que contenga, a razón de 10 fichas por línea, y presentando las fichas sin separación entre ellas. Deberá utilizar tablas en HTML para lograr el efecto de disponer varias fichas en una fila.

Todos los atributos han de ser privados, y todos los métodos han de ser públicos.

Para comprobar el funcionamiento de estas clases, escriba un programa que haga lo siguiente:

- Crear un montón con todas la fichas.
- Visualizarlo por consola
- Con un bucle, sacar 10 piezas de un montón y meterlas en otro, el cual estará inicialmente vacío.
- Visualizar ambos montones por consola

A continuación se muestra un posible código para probar algunos métodos:

Familia Profesional Informática y Telecomunicaciones		Nombre del Ciclo Formativo Título de Técnico Superior en Desarrollo de Aplicaciones Web			
Centro Educativo IES Campanillas (sede PTA)		Módulo Profesional Programación Código: 0485 N.º de créditos ECTS: 14		Profesor Juan Antonio Jiménez Morales	
Curso lectivo: 2019/2020	Grupo: 1º DAW	Trimestre: Terº – Control 8	Modelo: Único	Fecha: 21/05/2020	Pág. 3/8

```

public static void main (String[] args) {
    Monton miMonton1 = new Monton();
    Monton miMonton2 = new Monton();

    miMonton1.inicializaMonton();
    System.out.println("Montón 1 recién creado: ");
    System.out.println(miMonton1);
    for (int i=0; i<10; i++) {
        miMonton2.aniadeFicha(miMonton1.sacaFichaAlAzar());
    }
    System.out.println("Montón 1 después de sacar 10 fichas al azar: ");
    System.out.println(miMonton1);
    System.out.println("Montón 2 con las fichas que faltan en el montón 1: ");
    System.out.println(miMonton2);
}

```

Y el resultado de la ejecución podría ser (al ser extracciones aleatorias, no siempre obtenemos el mismo resultado):

```

Montón 1 recién creado:
[6|6] [6|5] [6|4] [6|3] [6|2] [6|1] [6| ] [5|5] [5|4] [5|3]
[5|2] [5|1] [5| ] [4|4] [4|3] [4|2] [4|1] [4| ] [3|3] [3|2]
[3|1] [3| ] [2|2] [2|1] [2| ] [1|1] [1| ] [ | ]
Montón 1 después de sacar 10 fichas al azar:
[6|6] [6|5] [6|4] [6|3] [5|5] [5|4] [5|3] [5| ] [4|3] [4| ]
[3|2] [3|1] [3| ] [2|1] [2| ] [1|1] [1| ] [ | ]
Montón 2 con las fichas que faltan en el montón 1:
[6|1] [4|1] [4|4] [4|2] [2|2] [5|1] [6| ] [6|2] [3|3] [5|2]

```

Familia Profesional Informática y Telecomunicaciones		Nombre del Ciclo Formativo Título de Técnico Superior en Desarrollo de Aplicaciones Web			
Centro Educativo IES Campanillas (sede PTA)		Módulo Profesional Programación Código: 0485 N.º de créditos ECTS: 14		Profesor Juan Antonio Jiménez Morales	
Curso lectivo: 2019/2020	Grupo: 1º DAW	Trimestre: Terº – Control 8	Modelo: Único	Fecha: 21/05/2020	Pág. 4/8

2. [3 puntos] Implemente la siguiente versión del “solitario” con fichas del dominó. Se trata de que el ordenador juegue “solo” al solitario. Para ello, dispondremos de todas las fichas del dominó en un montón “M” (podemos pensar que están boca abajo). El juego empieza eligiendo aleatoriamente una ficha de dominó. A continuación, y de manera repetitiva, se va eligiendo una ficha del montón “M”, y se ve si se puede colocar en alguno de los extremos de la “hilera” de fichas.

Si la ficha elegida no se ha podido colocar en ningún extremo de la “hilera”, se pasará a un montón auxiliar “A” y se volverá a elegir otra ficha del montón “M”. Una vez se consiga elegir una ficha del montón “M” que se pueda colocar en alguno de los extremos de la “hilera”, se devolverán todas las fichas del montón auxiliar “A” al montón “M”. Y se volverá a elegir otra ficha del montón “M”.

Por si no se había dado cuenta: de la colocación de la ficha en la hilera se ocupa un método de la clase Tablero. Y la hilera es un tablero.

El juego finaliza cuando se hayan colocado todas las fichas en la “hilera” o las fichas que queden en el montón auxiliar “A” no se puedan colocar en los extremos de la “hilera”. Dicho de otra manera, el juego finaliza cuando se agote el montón “M” de fichas (las que estaban boca abajo).

El programa debe mostrar en una página web o por consola el estado final en el que quedan la “hilera”, el montón auxiliar “A” y el montón “M” (el cual SIEMPRE deberá quedar vacío). Internamente, el programa ha de seguir los pasos descritos arriba.

Debe utilizar las clases y objetos del ejercicio 1. Se le facilitarán imágenes para generar las fichas de dominó. En el caso de utilizar el programa por consola, la puntuación máxima del ejercicio serán 2 puntos.

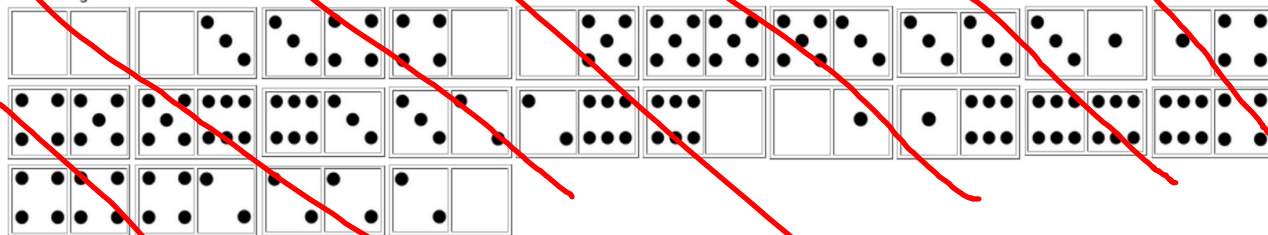
A continuación, se muestran posibles ejemplos de ejecución del programa que se pide (cada ejecución arroja resultados distintos):

Por consola:

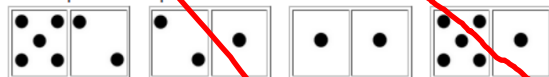
```
Solitario generado:
[1|3][3|3][3|2][2|2][2| ] [ | ] [6|6][6|4][4|2][2|1]
[1| ] [4|4][4|5][5|3][3| ] [5|5][5|2][2|6][6|5][5|5]
[5|1][1|6][6|3][3|4][4|1][1|1]
Fichas que no han podido colocarse:
[6|6] [4|4]
Fichas que quedan boca abajo:
```

Por página web:

Solitario generado:



Fichas que no han podido colocarse:



Fichas que quedan boca abajo:

Familia Profesional Informática y Telecomunicaciones		Nombre del Ciclo Formativo Título de Técnico Superior en Desarrollo de Aplicaciones Web			
Centro Educativo IES Campanillas (sede PTA)		Módulo Profesional Programación Código: 0485 N.º de créditos ECTS: 14		Profesor Juan Antonio Jiménez Morales	
Curso lectivo: 2019/2020	Grupo: 1º DAW	Trimestre: Terº – Control 8	Modelo: Único	Fecha: 21/05/2020	Pág. 7/8

4. [3 puntos] Como alternativa al ejercicio 2, puede optar por realizar el siguiente, siguiendo una metodología orientada a objetos, el siguiente ejercicio:

Los métodos que implemente no pueden realizar E/S por consola. Debe elegir qué clases implementar y con qué métodos y atributos para que tenga sentido hacerlo con esta metodología. Todo el código que incorpore en este ejercicio debe comprenderlo.

Implemente la generación de tickets de la compra al estilo "Mercadona". En el ticket de la compra de esta cadena de supermercados, aparece una línea por cada producto adquirido, indicándose:

Cantidad	Descripción	Precio por unidad	Importe
----------	-------------	-------------------	---------

Al final del listado de productos, se indica el precio total de la compra (con IVA incluido). El precio por unidad y el importe llevan el IVA incluido.

Bajo el precio total, aparece un detalle donde se desglosa, para cada tipo de IVA, la base imponible y la cuota correspondiente:

IVA	BASE IMPONIBLE	CUOTA
-----	----------------	-------

Hay 3 tipos (porcentajes) de IVA: 4%, 10% y 21%. Cada producto adquirido tiene un tipo de IVA asociado. La base imponible que aparece en el detalle del ticket es, para cada tipo de IVA, la suma de los importes (descontando el IVA) correspondientes a los productos que tienen un mismo tipo de IVA, y la cuota es la suma de los importes de IVA correspondientes a los productos que tienen un mismo tipo de IVA.

Debe implementar un programa que admita un máximo de 10 entradas interactivas de productos, permitiéndole finalizar la introducción de datos si elige el producto *fin*. **Para cada producto, el programa ha de solicitar: cantidad, descripción, precio por unidad (lleva el IVA incluido) y tipo de IVA.** A la hora de capturar el tipo de IVA, el programa debe asegurar que se elige uno de los tipos válidos.

Una vez capturados los datos, se debe mostrar el ticket de la compra. A continuación, se muestra un ejemplo de lo que podría ser la ejecución del programa. Tenga cuidado con presentar los datos correctamente alineados y formateados.

En lo referente al desglose de bases imponibles y de cuotas, y respecto al ejemplo que sigue, observe cómo hay dos productos con un IVA del 4%, un producto con un IVA del 10% y otro producto con un IVA del 21%. Los productos a1 y a4 tienen un coste conjunto de 17.00 €, lo que se desglosa en que de esos 17.00 €, 16.35 € corresponden al precio sin IVA y 0,65 € corresponden al impuesto (IVA 4%).

El segundo artículo tiene un IVA del 10%, y su coste asciende a 4.00 €, lo que se desglosa en que el coste sin IVA es de 3.64 € y el importe del IVA (del 10%) es de 0.36 €.

Y lo mismo, pero aplicando un porcentaje del 21%, ocurre para el coste acumulado de los productos cuyo tipo de IVA es del 21%. En el ejemplo, el coste acumulado de los productos que tienen ese IVA (sólo el tercer producto en el ejemplo) es de 10.50 €, de los cuales 8.68 € corresponden al coste sin el IVA y 1.82€ corresponden al impuesto (IVA 21%).

Al final del desglose, aparece la suma de todas las bases imponibles y la suma de todos los IVAs.

Familia Profesional Informática y Telecomunicaciones		Nombre del Ciclo Formativo Título de Técnico Superior en Desarrollo de Aplicaciones Web			
Centro Educativo IES Campanillas (sede PTA)		Módulo Profesional Programación Código: 0485 N.º de créditos ECTS: 14		Profesor Juan Antonio Jiménez Morales	
Curso lectivo: 2019/2020	Grupo: 1º DAW	Trimestre: Terº – Control 8	Modelo: Único	Fecha: 21/05/2020	Pág. 8/8

Introduzca nombre artículo ("fin" para salir): a1
 Introduzca la cantidad: 1
 Introduzca el precio por unidad: 1
 Introduzca el tipo de IVA (4, 10 o 21 %): 5
 Incorrecto. Introduzca el tipo de IVA (4, 10 o 21%): 4

Introduzca nombre artículo ("fin" para salir): a2
 Introduzca la cantidad: 2
 Introduzca el precio por unidad: 2
 Introduzca el tipo de IVA (4, 10 o 21 %): 10

Introduzca nombre artículo ("fin" para salir): a3
 Introduzca la cantidad: 3
 Introduzca el precio por unidad: 3.5
 Introduzca el tipo de IVA (4, 10 o 21 %): 21

Introduzca nombre artículo ("fin" para salir): a4
 Introduzca la cantidad: 4
 Introduzca el precio por unidad: 4
 Introduzca el tipo de IVA (4, 10 o 21 %): 4

Introduzca nombre artículo ("fin" para salir): fin

Cantidad	Descripción	Precio	Importe
1	a1	1.00	1.00
2	a2	2.00	4.00
3	a3	3.50	10.50
4	a4	4.00	16.00

		TOTAL	31.50
IVA	BASE IMPONIBLE	CUOTA	
4%	16.35	0.65	
10%	3.64	0.36	
21%	8.68	1.82	
TOTAL	28.66	2.84	