

EXÁMENES PRESENCIALES FP A DISTANCIA
FEBRERO 2022



CICLO FORMATIVO:
MÓDULO PROFESIONAL:
FECHA Y HORA:
DURACIÓN:

DAM
Programación
Martes 8 de febrero de 2022 15:30
2 horas y 45 minutos



EXAMEN:

Ejercicio 1. Crea un programa en Java para resolver el siguiente problema.

Un almacén de ropa almacena sus productos siguiendo un código formado por números y letras el cual permite reconocer el tipo de cliente al que van dirigidos, un identificador y su origen de fabricación. Su estructura es la siguiente:

- **Una letra** que indica el **tipo de cliente** (H - Hombre, M - Mujer o B - Bebé).
- Un código de **cinco dígitos numéricos** (**identificador**)
- **Una letra** que indica el **origen de fabricación** del producto: E (Europa) o A (Asia).
- **Dos dígitos de control**. Se formarán sumando todos los dígitos numéricos del código. Si la suma resultante es de una sola cifra se antepondrá el 0.

Las letras solo pueden ser mayúsculas. Si las letras son minúsculas el código se considera inválido.

Dados los siguientes códigos de producto: **H12345E15**, **M00000E00**, **A12345E15**, **H12345E11**, **B00011A02**, **B00011A11**, **h12345E15**, **B1111A04**, **B111111A06**, desarrolla un programa en Java que identifique cuántos productos tienen un patrón válido y, de ellos, cuántos corresponden a códigos válidos o a códigos inválidos. Por último, para los patrones válidos, identifica de qué tipo de cliente se trata (hombre, mujer, bebé) y su origen (Europa o Asia).

Puesto que no hay entrada de datos, la salida debe ser similar a la siguiente:

EJERCICIO DE ANÁLISIS DE CÓDIGOS DE PRODUCTOS TEXTILES

Se analizarán los siguientes códigos de producto:

[H12345E15, M00000E00, A12345E15, H12345E11, B00011A02, B00011A11, h12345E15, B1111A04, B111111A06]

RESULTADO

H12345E15: Prenda de hombre. Fabricación: Europa.
M00000E00: Prenda de mujer. Fabricación: Europa.
A12345E15: Patrón NO válido.
H12345E11: Código inválido.
B00011A02: Prenda de bebé. Fabricación: Asia.
B00011A11: Código inválido.
h12345E15: Patrón NO válido.
B1111A04: Patrón NO válido.
B111111A06: Patrón NO válido.

Número de productos con patrón válido:

5 productos (3 con código válido y 2 con código inválido)

EXÁMENES PRESENCIALES FP A DISTANCIA
FEBRERO 2022



CICLO FORMATIVO:
MÓDULO PROFESIONAL:
FECHA Y HORA:
DURACIÓN:

DAM

Programación

Martes 8 de febrero de 2022 15:30

2 horas y 45 minutos



Ejercicio 2

Dados dos arrays de diez elementos que se os proporcionan, escribir un programa en Java que cree un tercer array del mismo tamaño y con el siguiente contenido:

- En la primera y última posición se almacenarán, respectivamente, el menor y el mayor elemento que se encuentre entre los dos arrays.
- En el resto de posiciones, se almacenará la suma de los valores que haya en esa posición en cada array.
- Al final del programa, se debe mostrar el máximo de todos los elementos contenidos en este nuevo array.

Dado que los arrays se proporcionan y por tanto no hay entrada de datos, el resultado debería ser similar al siguiente:

```
EJERCICIO DE ARRAYS
```

```
-----
```

```
Array 1:  [1, 8, 3, 1, 3, 7, 5, 2, 4, 6]
```

```
Array 2:  [3, 4, 5, 6, 2, 5, 1, 7, 5, 5]
```

```
RESULTADO
```

```
-----
```

```
Array resultado: [1, 12, 8, 7, 5, 12, 6, 9, 9, 8]
```

```
Máximo del array resultado: 12
```

EXÁMENES PRESENCIALES FP A DISTANCIA
FEBRERO 2022



CICLO FORMATIVO:
MÓDULO PROFESIONAL:
FECHA Y HORA:
DURACIÓN:

DAM

Programación

Martes 8 de febrero de 2022 15:30

2 horas y 45 minutos



Ejercicio 3. Implementa una clase *Paciente* teniendo en cuenta las siguientes consideraciones.

Clase *Paciente*. Clase que modela un paciente considerando su **nombre**, **gasto acumulado** y **número de pinchazos** que lleva puestos en la clínica.

Te pedimos que implementes esta clase utilizando programación dirigida a objetos. Es decir, creando la clase que modele el comportamiento del paciente con el siguiente diseño:

Nombre de la clase: Paciente

Atributos de la clase:

- String nombre Nombre del paciente
- double gastoAcumulado Gasto acumulado del paciente
- int numPinchazos Número de pinchazos recibidos por el paciente

Método constructor: Paciente (String nombre, double gastoAcumulado, int numPinchazos). Este método instancia un objeto de la clase Paciente e inicializa los atributos con los valores que se pasan como parámetros. Si el gasto acumulado o el número de pinchazos es negativo, se lanza una excepción de tipo ***IllegalArgumentException*** con el mensaje "*Datos iniciales erróneos*"

Métodos de la clase:

```
boolean actualizarGastoAcumulado(int numPinchazosHoy, float  
importePinchazo)
```

Recibe como parámetros el número de pinchazos y el importe de cada uno. Si son valores positivos, entonces:

1. **se incrementa el número de pinchazos** según los recibidos como parámetro;
2. **se actualiza el gasto acumulado** incrementando el que se tenía teniendo en cuenta el importe de los pinchazos que ahora se ponen y el importe de cada uno.

Este método devuelve el valor **true** indicando que la operación se ha llevado a cabo correctamente. En otro caso se devuelve **false**.

EXÁMENES PRESENCIALES FP A DISTANCIA
FEBRERO 2022



CICLO FORMATIVO: DAM
MÓDULO PROFESIONAL: Programación
FECHA Y HORA: Martes 8 de febrero de 2022 15:30
DURACIÓN: 2 horas y 45 minutos



`String tipoPaciente()`.

Según el gasto acumulado del paciente, se devolverá alguno de los siguientes textos:

Premium	A partir de 10000 euros.
Muy interesante	A partir de 7750.
Interesante	A partir de 4000 euros.
Estándar	Hasta 4000 euros.

`String toString()`. Devuelve una cadena con el texto correspondiente al valor de los atributos, de modo que devolvería un *String*.

Ejemplo:

Paciente: Fulgencio Pérez, número de pinchazos: 6, gasto acumulado: 389,35

Método main de prueba de la clase

Crea también un método `main()` para probar la clase realizando las siguientes acciones:

1. Un primer intento de crear un objeto con datos erróneos para que salte la excepción y sea capturada como ya sabemos con una estructura *try-catch*.
2. Crear dos objetos *Paciente* correctos y mostrar su estado usando el método *toString*. Usar los datos indicados en el ejemplo.
3. Al paciente que has creado con el nombre de *Julian Piqueras Gómez*, **actualiza su gasto acumulado** empleando el método apropiado, con **dos pinchazos de 7025.50**. Muestra de nuevo por consola su información usando el método *toString*.
4. Al paciente que has creado con el nombre de *Fulgencio Pérez*, **actualiza su gasto acumulado** empleando el método apropiado, con **cinco pinchazos de 76.87**. Muestra de nuevo por consola su información usando el método *toString*.
5. Finalmente **muestra el nombre y tipo de paciente de ambos objetos** tal y como se ve en la ejecución siguiente, empleando los métodos *getNombre* y *tipoPaciente*.

**EXÁMENES PRESENCIALES FP A DISTANCIA
FEBRERO 2022**



CICLO FORMATIVO:
MÓDULO PROFESIONAL:
FECHA Y HORA:
DURACIÓN:

DAM
Programación
Martes 8 de febrero de 2022 15:30
2 horas y 45 minutos



Aquí tienes como debería quedar la salida de ejecución del programa de prueba una vez implementada la clase solicitada:

```
Intento crear un paciente de nombre Roberto
```

```
Error al intentar crear: Datos iniciales erróneos
```

```
Vamos a crear dos objetos paciente
```

```
Mostramos la información de los dos objetos paciente
```

```
Paciente: Julián Piqueras Gómez, número de pinchazos: 0, gasto acumulado: 0,00
```

```
Paciente: Fulgencio Pérez, número de pinchazos: 1, gasto acumulado: 5,00
```

```
Actualizamos el gasto acumulado de julian con dos pinchazos de 7025.50
```

```
Paciente: Julián Piqueras Gómez, número de pinchazos: 2, gasto acumulado: 14051,00
```

```
Actualizamos el gasto acumulado de Fulgencio con 5 pinchazos de 76.87
```

```
Paciente: Fulgencio Pérez, número de pinchazos: 6, gasto acumulado: 389,35
```

```
El tipo de paciente de: Julián Piqueras Gómez es Premium
```

```
El tipo de paciente de: Fulgencio Pérez es Estándar
```