

**EXÁMENES PRESENCIALES FP A DISTANCIA
FEBRERO 2022**

CICLO FORMATIVO: DAW
MÓDULO PROFESIONAL: Programación
FECHA Y HORA: Martes 8 de febrero de 2022 18:45
DURACIÓN: 2 horas y 45 minutos

Apellidos, Nombre:		DNI:
IES de referencia:	<input type="checkbox"/> IES Aguadulce (Almería) <input type="checkbox"/> IES Cristóbal de Monroy (Sevilla) <input type="checkbox"/> IES Trassierra (Córdoba)	
IES donde se realiza el examen:		

INSTRUCCIONES:

Selecciona tu centro de referencia y profesorado asignado:

IES Aguadulce	DAW	<input type="checkbox"/> José Javier	<input type="checkbox"/> Diosdado	<input type="checkbox"/> Sandra
IES Cristóbal de Monroy	DAW	<input type="checkbox"/> Ana	<input type="checkbox"/> Jesús	
IES Trassierra	DAW	<input type="checkbox"/> Fran	<input type="checkbox"/> José Antonio	

El examen práctico se puede realizar con todo el material que el alumnado estime oportuno (apuntes, libros, conexión a Internet y a la plataforma, pendrive, disco duro externo,...).

Se prohíbe el uso del móvil o de su propio portátil, debiendo hacerse con el ordenador que le facilita el centro, que debe disponer de conexión a Internet.

- El examen consta de 3 ejercicios.
- **Puntuación: 10 puntos en total.**

CALIFICACIÓN:

Mediante esta prueba se evalúan los resultados de aprendizaje y criterios de evaluación del módulo que se detallan a continuación:

RESULTADOS DE APRENDIZAJE	CRITERIOS DE EVALUACIÓN	PONDERACIÓN
RA1	a)	1%
	b)	1%
	c)	1%
	d)	1%
	e)	1%
	f)	1%
	g)	1%
	h)	1%
	i)	1%
RA2	a)	2%
	b)	1%
	c)	2%
	d)	2%
	e)	2%

**EXÁMENES PRESENCIALES FP A DISTANCIA
FEBRERO 2022**

CICLO FORMATIVO:

DAW

MÓDULO PROFESIONAL:

Programación

FECHA Y HORA:

Martes 8 de febrero de 2022 18:45

DURACIÓN:

2 horas y 45 minutos

	f)	2.5%
	g)	1%
	h)	2%
	i)	0.5%
RA3	a)	3%
	b)	3%
	c)	0.5%
	d)	2.5%
	e)	4%
	f)	2%
	g)	1%
RA4	a)	1%
	b)	3%
	c)	2%
	d)	1%
	e)	1%
	f)	1%
	h)	2%
RA5	a)	2%
	b)	1%
	c)	2%
RA6	a)	2%
	g)	1.75%
Peso total de la prueba en la calificación del módulo:		20 %

[3 puntos] Ejercicio 1. Crea un programa en Java para resolver el siguiente problema.

Una empresa de informática identifica sus productos siguiendo un código formado por números y letras el cual permite conocer el **tipo de producto**, su **identificador** y su **estado**. El código de los productos se construye de la siguiente manera:

- Una letra que indica la **disponibilidad** del producto: S (en stock) o A (agotado).
- Una letra que indica el **tipo de producto** (T - Tablet, P - Portátil o M - Móvil).
- Un **identificador** de **cuatro dígitos numéricos**.
- Un **código de control (una letra)**. Se formará aplicando la operación **módulo 10 a la suma de los cuatro dígitos numéricos del identificador**, por lo que se obtendrá una letra según el resultado de esta operación: 0 será la A, 1 será la B, etc. Hasta el 9, que será la J.

Las letras pueden ser **mayúsculas o minúsculas**. En ambos casos se trata de un código válido.

Dados los siguientes códigos de producto: **SP1234A**, **aM5544I**, **AO5925N**, **O26232A**, **AT5425E**, **SM4285J**, **sp1599A**, **SP12341B**, **SP111C**, desarrolla un programa en Java que identifique cuántos productos tienen un patrón válido y, de ellos, cuántos contienen códigos de control válidos y cuántos inválidos. Por último, para los patrones válidos, identifica de qué tipo de producto se trata (móvil, tablet o portátil) y si están Agotados o En stock.

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

EJERCICIO DE ANÁLISIS DE CÓDIGOS DE PRODUCTOS INFORMÁTICOS

Se analizarán los siguientes códigos de producto:

[SP1234A, aM5544I, AO5925N, O26232A, AT5425G, SM4285J, sp1599A, SP12341B, SP111C]

RESULTADO

SP1234A: Portátil (En stock)
aM5544I: Móvil (Agotado)
AO5925N: Patrón NO válido
O26232A: Patrón NO válido
AT5425G: Tablet (Agotado)
SM4285J: Móvil (En stock)
sp1599A: Código inválido
SP12341B: Patrón NO válido
SP111C: Patrón NO válido

Número de productos con patrón válido:

5 productos (4 con código de control válido y 1 inválido)

[3,5 puntos] 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 posición se almacenará la suma de todos los elementos del primer array y en la última posición se almacenará la suma de todos los elementos del segundo array.
- En el resto de posiciones, se almacenará el contenido de mayor valor de los arrays en esa posición.
- Al final del programa, se debe mostrar la suma 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: [40, 8, 5, 6, 3, 7, 5, 7, 5, 43]

Suma del array resultado: 129

[3,5 puntos] Ejercicio 3. Implementa una clase *Estudiante* teniendo en cuenta las siguientes consideraciones.

Clase *Estudiante*. Clase que modela un estudiante considerando su **nombre**, **nota media numérica real** y **número de notas** que se tienen en cuenta para calcular esa nota media.

Te pedimos que implementes una clase utilizando programación dirigida a objetos. Es decir, escribiendo el código de la clase que modele el comportamiento del estudiante con el siguiente diseño:

Nombre de la clase: Estudiante

Atributos de la clase:

- String nombre Nombre del estudiante
- double notaMedia Nota media del estudiante
- int numNotas Número de notas del estudiante

Tendrás que implementar un método get para cada atributo (getNombre, getNotaMedia, getNumNotas).

Método constructor: Estudiante (String nombre, double notaMedia, int numNotas). Este método instancia un objeto de la clase Estudiante e inicializa los atributos con los valores que se pasan como parámetros.

En caso de la nota media o el número de notas recibido sean menores que 0 lanza una excepción *IllegalArgumentException* indicando ***Datos iniciales erróneos***

Otros métodos de la clase:

`boolean actualizarNotaMedia(double nuevaNota)`. Si la nota que se recibe como parámetro no es negativa, se considera la actualización y se devolverá **true**. Así, se incrementará el número de notas de la persona y se actualizará la **nota media** con esta nueva nota que se recibe como parámetro. En otro caso, se devolverá **false**.

```
char calificacionMedia() .
```

Según el valor de la nota media, se devolverá alguno de los siguientes caracteres:

'A'	A partir de 9.
'B'	A partir de 8.
'C'	A partir de 7.
'D'	A partir de 6.
'E'	A partir de 5.
'F'	Menor de 5.

`String toString()` . Devuelve una cadena con el texto correspondiente al valor de los atributos, de modo que devolvería un `String`. como por ejemplo, supuesto un objeto con esos valores en sus atributos:

Ejemplo:

“Estudiante: Francisco Cucharas Gómez, nota media: 2.0, número de notas:1”

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 *Estudiante* correctos y mostrar su estado usando el método *toString*.
Usar los datos indicados en el ejemplo:
 - *Antonio Bernal Campillo*, nota media: 0,0, número de notas:0.
 - *Francisco Cucharas Gómez*, nota media: 2,0, número de notas: 1.
3. Al estudiante que has creado con el nombre *Antonio Bernal Campillo*, **actualiza su nota media con una nueva calificación de 7.7**. Muestra de nuevo por consola su información usando el método *toString*.
4. Al estudiante que has creado con el nombre *Francisco Cucharas Gómez*, **actualiza su nota media con una nueva calificación de 7.5**. Muestra de nuevo por consola su información usando el método *toString*.
5. Finalmente, **muestra por pantalla la calificación media de Antonio Bernal Campillo**.

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

```
Intento crear un estudiante de nombre Ricardo

Error al intentar crear: Datos iniciales erróneos


Vamos a crear dos objetos estudiante

Mostramos la información de los dos objetos estudiante

Estudiante: Antonio Bernal Campillo, nota media: 0.0, número de notas:0

Estudiante: Francisco Cucharas Gómez, nota media: 2.0, número de notas:1


Actualizamos la nota media de Antonio con una nueva nota de 7.7

Estudiante: Antonio Bernal Campillo, nota media: 7.7, número de notas:1


Actualizamos la nota media de Francisco con una nueva nota de 7.5

Estudiante: Francisco Cucharas Gómez, nota media: 4.75, número de notas:2


Mostramos la calificación media de Antonio

La calificación media de: Antonio Bernal Campillo es C
```