



Programación



Ejercicio final



# ENUNCIADO

## Ejercicio 1

Será necesario implementar tres clases **PruebaJugador**, **Jugador**, **Perfil**.

Un **Jugador** viene definido por su nombre (un String), su personaje (un String) y su perfil (objeto de la clase Perfil)

Un **Perfil** viene definido por tres datos: valor, carisma y suerte (todos ellos int)

El programa **PruebaJugador** será un programa básico con dos jugadores

## PruebaJugador.java

- En el programa se comienzan declarando dos jugadores j1 y j2.

Se crean los dos jugadores el primero, j1, de nombre "Juan" y personaje "enano". El segundo, j2, de nombre "Pedro" y de personaje "elfo". Los jugadores al crearse comienzan teniendo un perfil con los tres datos, valor, carisma y suerte con valor cero.

Se escriben en pantalla los datos del primer jugador y del segundo jugador:

```
System.out.println("Primer jugador " + j1);  
System.out.println("Segundo jugador " + j2);
```

- A continuación se cambia el perfil de los dos jugadores:

- Al jugador j1 se le asignarán los siguientes valores a su perfil; 3 en valor, 4 en carisma y 5 en suerte.
- Al jugador j2 se le asignarán los siguientes valores a su perfil; 5 en valor, 6 en carisma y 7 en suerte.

El cambio de perfil del jugador se realizará mediante un método de la clase Jugador de nombre **cambiarPerfil()**

Se escriben en pantalla los datos del primer jugador y del segundo jugador:

- Intercambiamos el personaje de los jugadores. j1 que tenía como personaje "enano" quiero que pase a ser "elfo" el personaje de j2. Y j2 pase a ser "enano" el personaje de j1.

---

Utilizamos el método de la clase Jugador de nombre **intercambiaraPersonajes(Jugador j)**

Se escriben en pantalla los datos del primer jugador y del segundo jugador:

- Se escribe en pantalla los datos del jugador ganador. Utilizando el método de la clase Jugador de nombre ganaAJugador(Jugador j)

Será más valorado el uso del operador condicional que una estructura if

### Perfil.java

---

La clase jugador como si indicó al principio tiene como atributos. valor (un int), carisma (un int) y suerte (un int)

El constructor tiene como parámetros tres int. Asigna el primero a valor, el segundo a carisma y el tercero a suerte.

El método toString ( ) devuelve un String con la información del Perfil en el siguiente formato:

```
Valor:  x
Carisma: y
Suerte: z
```

Un método nivelPerfil ( ) que devuelve un int con el dato la suma de los tres atributos (valor, carisma y suerte)

### Jugador.java

---

La clase jugador como se indicó al principio tiene como atributos. nombre (un String), su personaje (un String) y su perfil (objeto de la clase Perfil)

El constructor tiene como parámetros dos String. Asigna el primer String al atributo nombre y el segundo al atributo personaje. Cuando se crea un jugador el perfil comienza siendo 0 en todos los atributos (irá mejorando a medida que vaya jugando). Por lo tanto se crea un perfil con valores cero para la valor, carisma y suerte.

El método toString( ) devuelve un String con la información del Jugador en el siguiente formato:

```
nombre , personaje
Valor:  x
Carisma: y
Suerte: z
```

Un método cambiarPerfil( ) que recibe como parámetros tres números int. El método crea un nuevo Perfil con estos tres valores enteros para valor, carisma y suerte, y asigna este perfil al atributo perfil.

Un método ganaAJugador(Jugador j) devuelve un boolean si el primer un jugador gana a otro jugador dado como parámetro. Un jugador gana a otro jugador un perfil con un mayor nivel (suma de valor + carisma + suerte) utilizar método nivelPerfil( ).



---

## Ejercicio 2

---

Copiar un archivo con texto “palabras.txt” en un archivo “copia.txt”

---

## Ejercicio 3

---

Contar el número de líneas de un fichero de texto

---

## Ejercicio 4

---

Un programa que lea del fichero “datos.txt” que contiene nombres, peso y altura de personas (cada dato en una línea) y calcula el peso medio y la altura media.

---

## Ejercicio 5

---

Un programa más concreto que el anterior que permita leer del teclado los datos de peso, altura y nombre de la persona, los almacene en un array. Imprima en la pantalla el nombre de las personas de mayor y menor altura.

---

## Ejercicio 6

---

Modificar el programa anterior para que almacene y recupere los datos de un fichero.

---

## CRITERIOS DE CORRECCIÓN

---

Los criterios de corrección serán los siguientes:

Ejercicio1 : PruebaJugador	20%
Ejercicio 1: Perfil	15%
Ejercicio 1: Jugador	15%
Ejercicio 2:	10%
Ejercicio 3:	10%
Ejercicio 4:	10%
Ejercicio 5:	10%
Ejercicio 6	10%