

# Tarea para PROG03.

## Detalles de la tarea de esta unidad.

### Enunciado.

En esta unidad has visto las características fundamentales de la **programación orientada a objetos** y estudiado los **conceptos** más **importantes** de este modelo de programación. Has aprendido a crear y manipular **objetos**, qué son los **métodos** y cómo se definen los parámetros. También has podido conocer cómo se estructura la Biblioteca de Clases de Java, viendo algunas clases importantes, como por ejemplo las que nos permiten realizar la programación de la consola.

Para poder realizar la tarea de esta unidad vas a crear **una clase con una estructura básica**. Las tareas a realizar se centrarán en la creación de instancias de esa clase (objetos), la creación y utilización de unos métodos básicos, y el trabajo con constructores y parámetros.

Los Ejercicios van sobre la clase **Personaje** (estos ejercicios hazlos en un proyecto de Netbeans).

Además, se debe **crear un documento**, que  **acredite identidad**. Para ello basta con **algún pantallazo, del proyecto y ejercicios realizados y completos**, y que en el fondo salga vuestro Nombre y Apellidos dentro del aula virtual. Algo breve y conciso, sólo se trata de mostrar la identidad junto con vuestro trabajo, no de explicar paso a paso, ya que también me entregaréis el código para que pueda comprobar que funciona.

### Consideraciones sobre el código que implementes:

\* El código tiene que estar suficientemente comentado y bien indentado (también conocido como sangrado), si no, no se corregirá por ilegibilidad del mismo.

Además, hay que poner un comentario al principio de la clase **Personaje**, con tu nombre y apellidos, indicando el tema en el que estamos. Todo esto es requisito para continuar la corrección.

1. (1.5 puntos). Creación de proyecto, paquete, clase y recoger datos por teclado.

-Crea un proyecto y llámalo **ProyectoPersonajes\_NombreApellidos**. (donde NombreApellidos son los vuestros separados por guion bajo).

-Dentro de ese proyecto crea la clase **Personaje** que se define a continuación (la clase debe de estar dentro de un paquete que se llame **proyectopersonaje**):

```
package proyectopersonajes;

/*
 * @author ANTONIO
 */
public class Personaje {
    /*
     constantes que nos servirán para controlar el número máximo de algunas características
     de los personajes.*/
    final int MAX_FUERZA=100;
    final int MAX_INTELIGENCIA=100;
    //variable estática que se irá incrementando con la creación de cada personaje.
    static int numeroPersonajes;
    //nombre, edad, altura, inteligencia y fuerza del personaje
    String nombre;
    int edad; //en años
    int altura; // en centímetros
    int inteligencia;
    int fuerza;

    /*las 4 variables int podrian ser short pero es más didáctico el código resultante
    de esta manera para ahorrar conversiones casting*/

    String consultaNombre(){
        return this.nombre;
    }
    void cambiaNombre(String nombre){
        this.nombre=nombre;
    }
}
```

- A la clase implementada tienes que **añadirle** un método principal (main) que solicite una cadena al usuario por consola (*System.out.println()*), el usuario lo introduzca por teclado, y se asigne en el atributo **nombre**, a un objeto Personaje que hayas creado, para posteriormente mostrar por pantalla el nuevo valor del atributo.

**Ejemplo de ayuda:**

..... pides valor por pantalla y recoges el valor por teclado en una variable que se llame strNombre .....

.....

```
Personaje p1=new Personaje();
p1.cambiaNombre(strNombre);
```

..... a continuación muestras por consola p1.consultaNombre() .....

.....

## 2. (1.5 puntos) Métodos de consulta y asignación de las propiedades.

- Añade a la clase **Personaje** los métodos que faltan para poder **consultar** y **modificar** el valor del resto de los atributos (edad, altura, inteligencia y fuerza), de manera similar a como se han creado los métodos del atributo **nombre**, pero con sus características particulares (parámetros, tipos de datos, etc.).
- Después **completa el programa** (main y donde sea necesario) para comprobar el funcionamiento de los nuevos métodos, en el que se vea como asignas valores y los muestras, como se hizo en el apartado anterior apartado con el nombre.

## 3. (1 puntos) Creación de constructor por defecto.

Crea ahora un método **constructor** para la clase **Personaje**. Servirá para que al crear un objeto de tipo **Personaje** inicialice los siguientes valores fijos a sus atributos:

```
nombre="Desconocido"
altura=0
edad=0
inteligencia=0
fuerza=0
```

*\*Nota: A partir de aquí, si necesitas crear nuevos objetos, los objetos y los datos que asignes puedes ponerlos directamente en el código, sin necesidad de estar pidiendo nada por teclado.*

*Ejemplo:*

```
Personaje p= new Personaje();
p.cambiaNombre("Gandalf");
p.cambiaInteligencia(100);
.....
etc.
```

A continuación, desde el main, crea un objeto de tipo **Personaje** utilizando el constructor que acabas de implementar, para posteriormente mostrar el contenido de **todos sus atributos** por pantalla (saldrán sus valores por defecto).

Ejemplo: `System.out.println("El valor de la fuerza es: "+p2.fuerza");`

Ahora muestra de nuevo el valor de algún atributo, pero utilizando su método de **consulta** que creaste anteriormente.

Ejemplo: `System.out.println("El valor de la fuerza es: "+p2.consultaFuerza());`

## 4. (1 punto) Creación de constructor con parámetros.

Crea ahora un constructor con parámetros para la clase **Personaje** que inicialice los atributos del objeto con los valores que se pasen en los parámetros.

*Ejemplo de uso futuro de ese nuevo constructor, aunque en este apartado aún no hay que probarlo:*

```
Personaje p1=new Personaje ("Frodo", 33, 126, 98, 38);
```

En este constructor se inicializarán atributos con los valores pasados como parámetros, no como en el anterior constructor que era el constructor vacío e inicializa valores por defecto.

5. **(1.5 puntos)** Creación del toString().

Crea un método `public String toString()`: Convertirá y devolverá un `String` con todos los datos del personaje de tal manera que para un personaje concreto, se visualizaría así:

"Nombre: Frodo; Edad: 33 años; Altura: 126 cm; Inteligencia:98; Fuerza: 38"

6. **(1.5 puntos)** Creación del método sumarFuerza().

Crea un método `public int sumarFuerza(Personaje p);`

Este método recibirá un objeto de tipo `Personaje`, y aunará las fuerzas de ambos objetos de la siguiente manera:

Sumará la fuerza del objeto actual (`this`), con la fuerza del personaje que recibe como parámetro (`p`). El método devolverá esa suma. (aquí sólo hay que implementarlo, lo comprobaremos en el siguiente apartado)

7. **(2 puntos)** Comprobación de la comprensión y funcionamiento del programa y uso de constantes y variables estáticas.

*\* Como se indicó anteriormente, estos objetos y sus datos puedes ponerlos directamente en el código, no es necesario que estés pidiendo todo por teclado, ya que lo hiciste en el primer apartado. Para el resto de pruebas puedes crearlos con los valores que se vayan pidiendo, pero sin necesidad de pedirlos por teclado.*

A continuación, desde el `main`, crea *otro objeto* de tipo `Personaje` utilizando el **primer** constructor, para posteriormente mostrar el contenido de los atributos por pantalla, pero ahora utilizando el `toString()` creado. Se visualizaría así:

"Nombre: Desconocido ; Edad: 0 años; Altura: 0 cm; Inteligencia: 0; Fuerza: 0"

- Posteriormente tienes que crear *3 objetos* utilizando el **segundo** constructor, con los siguientes datos.

Personaje1: nombre="Frodo"; edad=33; altura=126; inteligencia=98; fuerza=38;

Personaje2: nombre="Aragorn"; edad=45; altura=185; inteligencia=88; fuerza=92;

Personaje3: nombre="Legolas"; edad=2931; altura=178; inteligencia=93; fuerza=84; (los elfos viven mucho no es un error)

- **Muestra** los datos de los personajes después de crearlos utilizando el `toString()`.

- Ahora **cambia** varias de sus propiedades utilizando los métodos que creaste en el apartado 2.

- **Vuelve a mostrar** los datos de los personajes para comprobar que han cambiado. (con `toString()`)

- Crea un **nuevo personaje inventado**, y muestra sus datos. (con `toString()`)

- **Cambia** sus datos y vuélvelo a mostrar. (con `toString()`)

- Cada vez que se crea un `Personaje`, puedes almacenarlos en diferentes variables (objetos).

- **Muestra por pantalla la suma de fuerzas** de dos objetos, usando el método `sumarFuerzas` de alguno de ellos (y pasarás el otro objeto como parámetro). Se visualizará algo como:

"La suma de la fuerza de **Frodo** y **Legolas** es: **XXX**" (dónde **XXX**, es el valor que devuelve el método, y los nombres serán los de los objetos elegidos)

- **Cada vez que se crea un personaje incrementarás en uno** el valor de la variable estática que creaste con la clase `Personaje`.

- **Antes de terminar, muestra el total de personajes que se llevan creados** (mostrando el valor de la variable estática)

- **Antes de terminar, muestra el valor de la constante `MAX_FUERZA`.**

Separa e indica con comentarios en el `main`, qué parte se refiere a cada uno de los apartados, para que al corregirlo pueda identificarlo.

### Criterios de puntuación. Total 10 puntos.

- Apartado 1: 1.5 puntos
- Apartado 2: 1.5 puntos
- Apartado 3: 1 puntos
- Apartado 4: 1 puntos
- Apartado 5: 1.5 puntos
- Apartado 6: 1.5 puntos
- Apartado 7: 2 puntos

**Total:** 10 puntos.

### Recursos necesarios para realizar la Tarea.

- Ordenador personal.
- Sistema operativo Windows o Linux.
- JDK y JRE de Java (preferiblemente últimas versiones)
- NetBeans IDE 6.9.1 o superior. (preferiblemente Apache NetBeans 12.5)

### Consejos y recomendaciones.

Dentro de la carpeta de NetBeans, se habrá creado automáticamente la carpeta del proyecto. Comprueba que efectivamente está la carpeta y dentro todo el contenido que has creado. Para entregar, comprime en un único archivo la carpeta del proyecto con todo su contenido, junto con el documento y pantallazos que demuestran la identidad.

### Indicaciones de entrega.

Se enviará **un único archivo comprimido** que contendrá el **proyecto**, y un **documento** PDF, con algún **pantallazo** que **muestre** los ejercicios **realizados y completos**, junto con vuestra identidad, poniendo de fondo por ejemplo la pantalla del aula virtual. (no se trata de explicar paso a paso, sino, ver que está todo hecho, y se ve vuestra identidad del aula virtual en los pantallazos).

El envío se realizará a través de la plataforma y el archivo se nombrará siguiendo las siguientes pautas: **apellido1\_apellido2\_nombre\_SIGxx\_Tarea**

Asegúrate que el nombre no contenga la letra ñ, tildes ni caracteres especiales extraños. Así por ejemplo la alumna **Begoña Sánchez Mañas para la tercera unidad del MP de PROG**, debería nombrar esta tarea como...

**sanchez\_manas\_begona\_PROG03\_Tarea.**