

PROGRAMACIÓN FRONTEND Y BACKEND

Java 8

Introducción a la POO (EJERCICIOS)

Edición: Junio 2021

1. Crea una clase llamada **Persona** con 2 propiedades: nombre y edad, con sus setters y getters. Se debe comprobar que la edad no es negativa para poder cambiarla. Crea también un constructor parametrizado que reciba el nombre y la edad y se los asigne.

Crea 1 objeto de la clase **Persona**. Intenta ponerle una edad negativa para comprobar que no cambia. A continuación muestra por la consola lo siguiente con los datos de la persona (esto es un ejemplo):

Pepito tiene 16 años

2. Crea un método llamado **saluda** en la clase **Persona** que no reciba ni devuelva nada. En ese método muestra el mensaje del ejercicio anterior por la consola. Modifica el **main** para que llame a dicho método en lugar de mostrar el mensaje directamente.
3. Crea un constructor de copia en la clase **Persona**. Pruébalo generando otra persona a partir del objeto creado en el ejercicio 1, cambiando el nombre y la edad y comprobando que no cambian en la persona original (prueba primero a hacerlo usando una asignación: $\text{Persona } p2 = p1$; para ver como realmente así no estás copiando el objeto).
4. Crea un array con 4 personas. Recórrelo y llama al método **saluda** en cada persona.
5. Crea un método estático llamado **copiaArray**, que reciba un array de personas y te devuelva otro array con las mismas posiciones y copias de las personas del primer array.

En el **main**, crea una copia del array del ejercicio anterior, modificando las propiedades de alguna persona (la posición que tú quieras). Después, haz que la persona en la misma posición del otro array salude para comprobar que no has modificado el objeto original sino la copia generada.

6. En el array de personas introduce algunas posiciones a null (prueba con posiciones intermedias). Observa que ahora el programa dejará de funcionar al intentar la copia del array (cuando intente acceder a las propiedades de un objeto nulo para copiarlas en el constructor de copia). Modifica el método **CopiaArray** para que compruebe y tenga en cuenta que una posición puede ser null (en cuyo caso asigna también null en lugar de intentar copiar el objeto).
7. Crea un método en la clase **Persona** que se llame **estoyEnArray**, que reciba un array de personas por parámetro y devuelva un booleano indicando si el objeto actual (**this**) está en el array o no.

Para probarlo, guarda las personas en variables y después mete dichos objetos en el array (pero no todos). Prueba a buscar una persona en un array con el método que hemos creado y muestra un mensaje indicando si se encuentra o no en el array.

8. Crea una clase llamada **Empresa** que tenga las propiedades **nombre** y **empleados** (array de Personas). Utiliza también una propiedad llamada **numEmpleados** para llevar la cuenta de cuántos empleados tiene la empresa.

Crea un constructor que reciba el nombre de la empresa y lo asigne. Además, deberá inicializar el array de empleados con 100 posiciones, y la propiedad **numEmpleados** a cero. Crea una empresa y muestra "He creado la empresa (nombre de la empresa)" después por la consola.

9. Crea el método **contrataEmpleado** en la clase **Empresa**. Este método recibirá un objeto de tipo **Persona** y lo añadirá al final del array.

En la empresa del ejercicio anterior, contrata 3 empleados. Después recorre los empleados y muestra sus datos por pantalla.

10. Crea 2 métodos llamado **despideEmpleado** en la clase **Empresa**. En el primero se recibirá la posición del array de empleados a borrar (**entero**), y en el segundo el nombre del empleado a borrar (**string**), por lo que será un método sobrecargado. Ambas versiones deben borrar del array de empleados a la persona correspondiente.

Prueba el método, haciendo que la empresa contrate a más empleados (que al menos tenga 5 o 6 en total) y borrando uno por posición y otro por nombre. Muestra los empleados de la empresa para comprobar que se han borrado correctamente los que tocaban.