



DWEC 2º DAW

UNIDAD 3: PROGRAMACIÓN CON OBJETOS

Curso 2.024-2025

PRÁCTICA OBJETOS Y CLASES

1. Crea un programa que defina un **objeto** a través de object literal. Lo que queremos crear es un **intervalo de números**. Acciones que realizar con el objeto:
 - a) El objeto tiene dos propiedades: número mínimo y máximo. Los valores necesarios los pediremos al usuario y haremos las comprobaciones necesarias para realizar el ejercicio correctamente.
 - b) Definiremos una propiedad calculada con “get” que generará un array con los números comprendidos entre el mínimo y el máximo de sus propiedades.
 - c) Mostraremos en la pantalla tanto el array como el valor mínimo y máximo

El array según su intervalo es: 3,4,5,6,7,8,9,10

El valor mínimo del array es: 3

El valor máximo del array es: 10

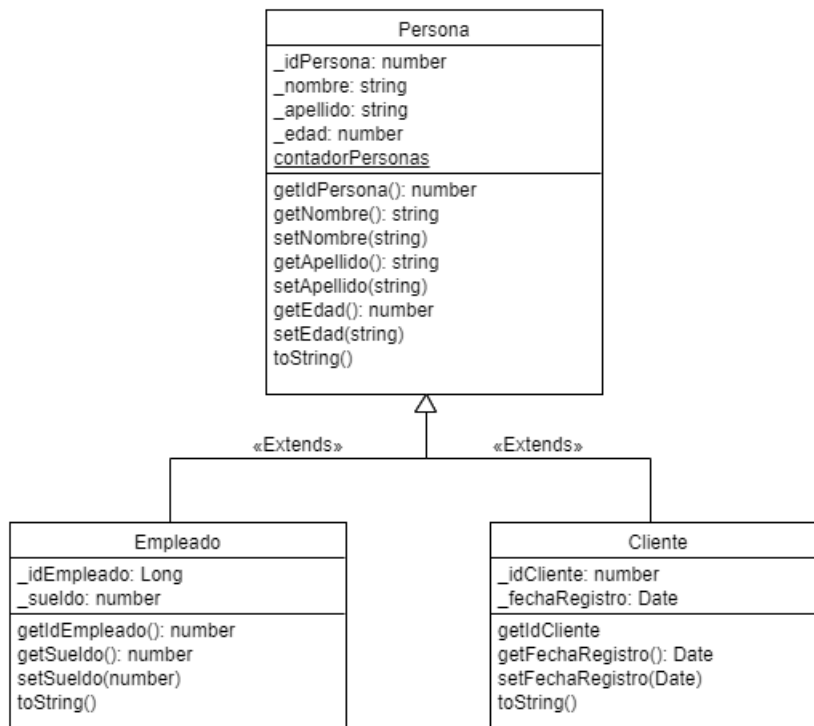
- d) Definiremos una propiedad con “set” para modificar el valor de las propiedades de nuestro objeto. A través de esa propiedad (que recibirá como argumento un array de 5 números aleatorios del 1 al 100), modificaremos los valores de mínimo y máximo de nuestro objeto
- e) Mostraremos por pantalla el nuevo array y los nuevos valores mínimo y máximo

El array con números aleatorios es: 92,27,46,66,16

El valor mínimo del nuevo array es: 16

El valor máximo del nuevo array es: 92

2. Crea las siguientes **clases** con su herencia. Te basarás en el siguiente diagrama UML:



Encontramos la clase Persona y las clases Empleado y Cliente que son “hijas” de Persona.

Respecto a la **clase Persona**:

- Se crearán los atributos indicados y uno estático que será contadorPersonas
- En el constructor de la clase se asignará al atributo _idPersona el valor del atributo estático incrementado en cada creación de un objeto. Es decir, ese valor NO me lo facilitarán cada vez que se cree un objeto, sino que será autocalculado. El constructor solo tendrá 3 parámetros.
- En cuanto a los **métodos**:
 - getIdPersona()→ servirá para extraer la propiedad _idPersona que NO se podrá modificar.
 - get y set para el atributo nombre y lo mismo para apellido y edad.
 - Se modificará el método toString() de la clase Object para que muestre la información completa de cada objeto.

Respecto a la **clase Empleado**:

- Crear un contador solo para empleados (similar al de personas)
- En cuanto a los **métodos**:
 - getIdEmpleado()→ servirá para extraer la propiedad que NO se podrá modificar.
 - get y set para el atributo sueldo.
 - Se reutilizará y modificará el método toString() de la clase Persona para que muestre la información completa de cada objeto de clase Empleado.

Respecto a la **clase Cliente**:

- Crear un contador solo para clientes (similar al de empleados)
- En cuanto a los **métodos**:
 - getIdCliente()→ servirá para extraer la propiedad que NO se podrá modificar.
 - get y set para el atributo fechaRegistro.
 - Se reutilizará y modificará el método toString() de la clase Persona para que muestre la información completa de cada objeto.

Crearemos y mostraremos por consola 6 objetos (2 de tipo persona, 2 de tipo empleado y 2 de tipo cliente) con la información completa de los métodos toString(), viéndose los id de personas y, a su vez, los

id de cada tipo de objeto (empleados y clientes).

Ejemplo de salida:

```
1: Carmen García, 65 años
2: Carlos Pérez, 25 años
3: Laura González, 28 años
   Empleado/a 1: 1500€
probando setter
4: Pietro Sánchez, 32 años
   Empleado/a 2: 1200€
5: Rodrigo Márquez, 50 años
   Cliente/a: 1 Fecha alta: Wed Sep 25 2024
6: Pietro Sánchez, 32 años
   Cliente/a: 2 Fecha alta: Wed Sep 25 2024
```