









# Tarea 05 - Diseño de la clase Persona

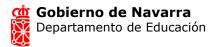
Todas las clases incluidas en los proyectos de los ejercicios que haya que entregar estarán debidamente comentadas incluyendo un comentario de clase y un comentario de método (comentarios **javadoc**). En el comentario de la clase además de una breve descripción acerca de la clase se incluirá el autor (@author) y la versión (@version)

Se respetará escrupulosamente las reglas de estilo del lenguaje Java (además de los comentarios, reglas de indentación, nombres descriptivos para variables y métodos, capitalización de nombres, blancos de separación entre los operadores, ...).

### Apartado 1: Creación de la clase

En esta primera parte realiza los dos siguientes pasos:

- Implementa la clase *Persona* que sea capaz de almacenar información sobre DNI, nombre, apellidos, sexo, fecha de nacimiento, Si tiene carnet de conducir, nacionalidad y dirección. Los atributos deberán estar declarados de manera que solo puedan verse por la propia clase *Persona*.
- Añade a la declaración de la clase *Persona* los métodos apropiados para lectura y escritura de los campos y establece los niveles de visibilidad adecuados a los miembros.







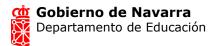


# Apartado 2: Constructores y método estático

- Declara el método constructor necesario para crear un objeto de la clase Persona pasando como argumento los datos de todos los campos.
- Declara otro constructor que inicialice los valores a un valor por defecto y que no reciba ningún parámetro.
- Incluye en la clase un campo estático que almacene el número de instancias creadas de la clase (deberá incrementarse en cada ejecución del método constructor).

## Apartado 3: Método con parámetros

- Añade a la clase *Persona* el método *getInfo* que retorne la información del Persona. Deberá ser declarado con el parámetro *formato*. Este parámetro admitirá dos valores:
  - "pausado": Se irá mostrando por pantalla la información de todos los campos y esperará a recibir un intro por parte del usuario para seguir mostrando los siguientes.
  - \* "normal": Se mostrará toda la información en un único paso.









 Añade a la clase *Persona* el método *compararEdad* que compare la edad del objeto persona con otra *Persona* e indique si la edad es igual, menor o mayor que el parámetro pasado. Además, añade dos métodos sobrecargando al método *compararEdad* que comparen la edad con un número y con una fecha.

```
compararEdad(Persona p1){ }
compararEdad(localDate fecha){ }
compararEdad(int año){ }
```

 Añade a la clase *Persona* un método sobrecargado adicional compararEdad que compare la edad del objeto persona con un String en el formato "ddmmaaaa", es decir, por ejemplo "23061998".

```
//Ejemplo de la llamada a la función compararEdad con el parámetro String
personal.compararEdad("23061998");
```

# Apartado 4: Método Main

 Añade una nueva clase *Principal* con el método main, que pruebe la ejecución de todos los métodos creados en la clase *Persona*. Crea al menos 2 objetos de la clase Persona para ello.









#### Criterios de puntuación. Total 10 puntos

- Apartado 1: 1 puntos.
- Apartado 2: 3 puntos.
- Apartado 3: 4 puntos.
- Apartado 4: 1 puntos.
- Comentarios y estructuración del código: 1 punto.
- Total: 10 puntos.

### Recursos necesarios para realizar la Tarea.

- Ordenador personal.
- JDK y JRE 21 o superior.
- NetBeans IDE 19 o superior.

### Consejos y recomendaciones.

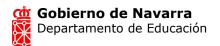
- Añade al código fuente los comentarios que consideres oportuno.
- Estructura el código de una forma correcta para un mejor seguimiento y corrección.

#### Indicaciones de entrega

El ejercicio estará contenido en un fichero cuyo nombre sea similar a **PROG05\_Ejerc01.** Se debe entregar el proyecto Netbeans creado. Para empaquetar un proyecto en Netbeans, utiliza la opción **File - Export Project** de Netbeans: generarás un fichero .zip con el contenido completo del proyecto.

Asegúrate de que los archivos fuente (.java), contengan, como comentarios Java, tu nombre y apellidos y el número de ejercicio.

Los archivos fuente deben compilar y ejecutarse siguiendo las instrucciones de su enunciado. De no hacerlo, no se considerarán válidos y serán evaluados con un 0.









En el supuesto que tengas/quieras adjuntar un documento de texto con explicaciones adicionales, o con la solución a alguno de los puntos pedidos, debes de seguir las siguientes consideraciones que se explican en el siguiente documento.

Crea una carpeta para la entrega y adjunta los archivos creados, y renombra la carpeta siguiendo las siguientes pautas:

apellido1\_apellido2\_nombre\_PROG\_Tarea05

Asegúrate que el nombre no contenga la letra ñ, tildes ni caracteres especiales extraños. Así por ejemplo la alumna Begoña Sánchez Mañas debería nombrar esta tarea como...

sanchez manas begona PROG Tarea05

## **IMPORTANTE**

Si la carpeta es muy pesada para el límite máximo de subida de la plataforma Moodle, súbela a tu unidad del Drive. Utiliza la cuenta de Educación, y en la entrega en Moodle adjunta la URL de tu entrega.

Para que la entrega se considere realizada en tiempo y forma, sube al Moodle un documento con la URL que contenga la solución de tu tarea. Deberá contener los archivos y documentos solicitados. Comparte la carpeta mediante su enlace correspondiente al Drive.

No olvides ajustar los privilegios de la carpeta en Drive para que pueda acceder a tu trabajo.

Todos los ejercicios y/o /documentos demandados que no cumplan exactamente con las instrucciones y su formato de entrega no se considerarán válidos y serán evaluados con un 0.