

UD3 – Prueba de validación – Calculadora y mascotas**Resultados de aprendizaje:****2. Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.****Criterios de evaluación:**

- a Se han identificado los fundamentos de la programación orientada a objetos.
- b Se han escrito programas simples.
- c Se han instanciado objetos a partir de clases predefinidas.
- d Se han utilizado métodos y propiedades de los objetos.
- e Se han escrito llamadas a métodos estáticos.
- f Se han utilizado parámetros en la llamada a métodos.
- h Se han utilizado constructores.
- i Se ha utilizado el entorno integrado de desarrollo en la creación y compilación de programas simples.

4. Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos.**Criterios de evaluación:**

- a Se ha reconocido la sintaxis, estructura y componentes típicos de una clase.
- b Se han definido clases.
- c Se han definido propiedades y métodos.
- d Se han creado constructores.
- e Se han desarrollado programas que instancien y utilicen objetos de las clases creadas anteriormente.
- f Se han utilizado mecanismos para controlar la visibilidad de las clases y de sus miembros.
- g Se han definido y utilizado clases heredadas.
- h Se han creado y utilizado métodos estáticos.
- j Se han creado y utilizado conjuntos y librerías de clases.

Recuerda:

- El examen finalizará a las 11:45, no se corregirá ninguna prueba entregada después de la hora marcada en la plataforma de elearning.
- Una vez finalizada la prueba, avisa al profesor para que la corrija contigo.
- Durante la prueba podrás consultar todas las fuentes que necesites, tanto apuntes, como prácticas o internet. Sin embargo, queda totalmente prohibido el uso de sistemas de mensajería instantánea como Whatsapp, Telegram, Slack, Twitch, etc...
- Durante la prueba no se podrá salir del aula.
- La presencia y uso de dispositivos móviles, ya sean estos tablets, teléfonos o smartwatches está totalmente prohibida.
- Para corregir esta práctica, se utilizará la rúbrica registrada en la plataforma de elearning.
- **Las actividades deben realizarse en un solo proyecto. Cada actividad tendrá su paquete principal con su propia clase ejecutable.**

Primer problema

Crea una clase en Java que disponga de los siguientes tres métodos estáticos:

a) **factorialDeN**: Debe recibir un parámetro que será el valor de N y debe devolver el resultado de multiplicar todos los números enteros comprendidos entre 1 y N, ambos incluidos.

Por ejemplo: Si el método recibe como parámetro 6, debe devolver el resultado de la siguiente operación: $1 * 2 * 3 * 4 * 5 * 6$

b) **mediaHastaN**: Debe recibir un parámetro positivo que será el valor de N y debe devolver el resultado de calcular la media de los números enteros comprendidos entre 1 y N, ambos incluidos.

Por ejemplo: Si el método recibe como parámetro 6, debe devolver el resultado de la siguiente operación: $(1 + 2 + 3 + 4 + 5 + 6) / 6$

c) **obtenerPasosFibonacci**: Debe recibir un parámetro positivo N y debe devolver el resultado de los N primeros pasos de la [sucesión de Fibonacci](#).

Por ejemplo: Si el método recibe como parámetro 7, debe devolver una cadena con los siete primeros pasos de la sucesión de Fibonacci: "0, 1, 1, 2, 3, 5, 8".

El programa principal de la clase ejecutable debe solicitar al usuario un valor numérico N y, a continuación, mostrar la salida de los tres métodos para ese valor.

Segundo problema

Crea una clase que represente a un **vehículo** y cumpla con estas características:

a) Debe contener los siguientes atributos:

- Marca
- Peso
- Potencia (en kW)
- Número de puertas
- Velocidad (en km/h)

b) Debe disponer de un constructor y de todos los métodos de acceso.

c) Debe disponer de un método que le permita acelerar. Este método debe sumar 10 unidades al atributo velocidad del vehículo.

A continuación, crea dos clases **ciclomotor** y **camión** que cumplan con estos requisitos:

a) Estas clases deben heredar de la clase vehículo anterior.

- El ciclomotor debe tener un atributo adicional que haga referencia a si está habilitado para ir por la autopista.

- El camión debe tener un atributo adicional que haga referencia a su capacidad de carga.
- b) Deben sobrecargar el método que permite acelerar a los objetos vehículo.
- En el caso de la clase ciclomotor, este método debe sumar 7 unidades a su atributo velocidad.
 - En el caso de la clase camión, este método debe sumar 5 unidades a su atributo velocidad.
- c) Sobrecarga el método toString de ambas clases de manera que generen una salida idéntica a las siguientes:

Ciclomotor	----- * Tipo de vehículo: Ciclomotor. * Marca: <Mostrar el valor del atributo correspondiente> * Peso: <Mostrar el valor del peso> * Potencia: <Mostrar el valor del atributo correspondiente> * Número de puertas: <Debe mostrarse aquí el valor correspondiente> * Puede ir por autopista: <Debe mostrarse aquí "Si" o "No"> * Velocidad actual: <Mostrar el valor correspondiente con unidades> -----
Camión	----- * Tipo de vehículo: Camión. * Marca: <Mostrar el valor del atributo correspondiente> * Peso: <Mostrar el valor del peso> * Potencia: <Mostrar el valor del atributo correspondiente con unidades> * Número de puertas: <Debe mostrarse aquí el valor correspondiente> * Capacidad de carga: <Debe mostrarse aquí el valor correspondiente> * Velocidad actual: <Mostrar el valor correspondiente con unidades> -----

Por último, en la clase ejecutable, instancia un objeto de cada tipo, haz que aceleren y muéstralos por pantalla empleando el comando sout.