

Tema 0

Ejercicios de introducción

Funcionamiento básico de Java y Estructuras de control

1. Escribe una clase en Java que represente una Calculadora. Considera que solo se pueden realizar operaciones con dos operadores. Añade los siguientes métodos:
 - Getter y setter para los dos operadores. Los **getters** devuelven el valor actual de un atributo, y los **setters** permiten establecerlo a través de un valor recibido por parámetros.
 - Suma de operadores.
 - Resta de operadores.
 - Multiplicación de operadores.
 - División de operadores.
 - Resto de la división entre operadores.
 - Mostrar por pantalla el contenido del operador 1.
 - Mostrar por pantalla el contenido del operador 2.
 - Raíz cuadrada del operador 1.
 - Raíz cuadrada del operador 2.
 - ¿Puedes implementar el cálculo de la potencia del primer operando elevado al segundo sin utilizar bibliotecas externas?

2. Escribe una clase en Java que represente un Libro. Cada libro tiene un título, un autor, un nº de páginas, y un género. Implementa las siguientes **restricciones** y **comportamientos** a través de métodos:

- El título puede contener espacios, pero no puede tener más de veinte caracteres.
- El nombre del autor no puede contener números.
- El nº de páginas debe ser válido: menos de 10 páginas se consideran Folletos, y más de 2000 páginas se consideran Enciclopedias.
- El género no puede contener espacios, y debe presentar todos los caracteres en minúsculas.
- Setters y getters para todos los atributos.
- Comprobar si un autor recibido por parámetros es el autor del libro actual.
- Comprobar si el título recibido por parámetros es el autor del libro actual.
- Comprobar si el libro actual tiene el mismo o menor número de páginas que el libro actual.
- Devolver un mensaje customizado dependiendo del género del libro:
 - Si el libro es de Fantasía: “Prepárate para vivir una aventura fantástica.”
 - Si el libro es de Ciencia Ficción: “Disfrutarás de una aventura más allá de tus expectativas.”
 - Si el libro es de Misterio: “No serás capaz de soltarlo.”
 - Si el libro es de cualquier otro género: “Disfruta de la gran variedad de nuestra biblioteca.”

3. Considera los siguientes bucles. Determina cuales son bucles infinitos, y por qué.

```
for(int i=1; i<11; i+=2){
    i--;
    System.out.println(i);
}

for(int i=1; i!=10; i++){
    i++;
    System.out.println(i);
}

for(int i=0; i<=100; i*=2){
    System.out.println(i);
}

for(int i=1; i<=100; i*=2){
    System.out.println(i);
}

for(int i=0; i%2==0; i+=2){
    if(i==50){
        i++;
    }
    System.out.println(i);
}
```

4. Escribe una función que reciba un número entero por parámetros, y calcule el factorial de ese número. El factorial, representado como $n!$, es como sigue:

$$n! = n*(n-1)*(n-2)...(n-(n-1))*1.$$

Por ejemplo, $5! = 5*4*3*2*10$

5. Escribe una función que reciba por parámetros un número positivo, y calcule cuantas cifras tiene dicho número. Asegúrate de que el parámetro recibido haya transmitido un número válido.

Programación orientada a objetos: conceptos avanzados

6. La clínica veterinaria Furball quiere automatizar los datos de todos los animales que pasan por la clínica. Siempre toman, como mínimo, los siguientes datos: nombre de la mascota, nombre del dueño, tipo de animal y el historial clínico. También quieren aprovechar para llevar un recuento de los perros y gatos que van pasando para hacer una estadística. Escribe una clase bajo las siguientes condiciones:
 - El nombre de la mascota siempre se escribe todo en mayúsculas.
 - El nombre del dueño se escribe sin espacios.
 - El historial clínico no puede exceder los 500 caracteres.
 - El tipo de animal solo puede ser perro, gato, o exótico.
 - El recuento de perros y gatos se lleva a cabo por separado y se hace a nivel de toda la clínica.
 - Añade un constructor por defecto, uno con parámetros que obligue a que se cumplan todos los requisitos, y uno de copia.
 - Añade getters y setters. Los setter también deberán obligar a que se cumplan las condiciones.
 - Reescribe los métodos clone(), equals() y toString() para esta clase.

7. Escribe una clase CentroEducativo que permite almacenar la información relativa a un centro de estas características. Todo centro se identifica mediante un nombre, el número de estudiantes, el número de profesores, y el código postal asociado.
 - El nombre del centro tiene un mínimo de 10 caracteres y un máximo de 30.
 - El código postal tiene hasta cinco dígitos, y el valor máximo de dicho código es de 52999.
 - El número de estudiantes no puede ser menor de 50 ni mayor de 1200.
 - El número de profesores no puede ser menor de 5 ni mayor de 200.
 - Añade un constructor por defecto y un constructor de copia.
 - Añade un constructor con parámetros que considere todas las restricciones establecidas.
 - Añade getters y setters.
 - Añade los métodos toString(), clone(CentroEducativo) e equals(CentroEducativo).

8. Escribe las siguientes clases que hereden de CentroEducativo:

- ColegioPrimaria: los colegios de Educación Primaria tienen una plantilla de como máximo 600 alumnos y 100 profesores. Además, tienen un evento temático mensual, que se anuncia al centro todos los meses. En ocasiones, se realiza algún evento complementario, de forma que en el anuncio mensual se notifican los dos eventos al mismo tiempo.
- InstitutoSecundaria: los institutos de Educación Secundaria tienen las restricciones de plantilla estándar, pero se agrupan en categorías: A hasta 400 estudiantes, B entre 400 y 700 estudiantes, C entre 700 y 1000 estudiante, y D entre 1000 y 1200. Suelen realizar competiciones que incluyen a grupos de entre 1 y 30 estudiantes, anunciando la cantidad de alumnos que van a participar. En ocasiones, otros Institutos participan también en la competición, y se anuncia también cuántos alumnos participan de otros centros.
- CentroFP: los centros de Formación Profesional tienen además una empresa asociada para que sus estudiantes realicen prácticas. En ocasiones, otras empresas distintas de las asociadas pueden permitir al alumnado realizar prácticas con ellos.
- Añade constructores por defecto, con parámetros y de copia a las subclases.
- Añade getters y setters cuando sea necesario.
- Añade los métodos toString(), clone() e equals().
- Lista todos los métodos sobrescritos y todos los métodos sobrecargados que has implementado.

Estructuras de datos

9. Considera a una Persona definida por los siguientes atributos: dni, apellidos, nombres, sexo, edad y peso. Implementa una clase que permita realizar esa representación, con constructores (defecto, parámetros, copia), getters, setters y toString(). Luego, crea un programa que permita registrar en un ArrayList una lista de Personas, y que admita las siguientes funcionalidades:
- Buscar a una Persona dado el dni.
 - Eliminar a una Persona según su nombre y apellidos.
 - Ordenar la lista de Personas por edad.
 - Mostrar todas las Personas almacenadas.
10. Crea las clases Empleado y Estudiante que hereden de Persona. Un Empleado tiene un sueldo, y un Estudiante un nivel de enseñanza. Añade getters y setters para los nuevos atributos y sobrescribe aquellos métodos que consideres necesarios. Después, intenta añadir a la clase del ejercicio anterior objetos de estas dos clases. ¿Qué observas?
11. Considera una Agenda telefónica como un conjunto de números de teléfono con nombres asociados. Utilizando un HashMap, implementa una clase que permita representar dicha Agenda con las siguientes funcionalidades:
- Comprobar si la Agenda está actualmente vacía.
 - Obtener el número de contactos insertados en la Agenda.
 - Añadir un contacto a la Agenda, consistente de su número de teléfono y el nombre asociado.
 - Confirmar si un contacto está en la Agenda, bien por su número de teléfono o por su nombre asociado.
 - Eliminar un contacto según su número de teléfono.
 - Modificar el nombre asociado de un teléfono concreto.
 - Modificar el nombre asociado de un teléfono concreto a partir del nombre asociado previo.
 - Vaciar la Agenda.

